

Программная инженерия

Пр 2
2015
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Михайленко Б.Г., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Липаев В.В., д.т.н., проф.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.С., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана, ОАО "Концерн "Сириус"

СОДЕРЖАНИЕ

- Корзун Д. Ж.** Формализм сервисов и архитектурные абстракции для программных приложений интеллектуальных пространств 3
- Большаков А. А., Лачугин Д. В., Лобанов В. В.** Проектирование программного комплекса адаптивной системы управления, сбора и анализа потоковых данных 13
- Жаринов И. О., Жаринов О. О.** Способ программного управления яркостью бортового средства индикации в режиме ночного полета летательного аппарата 23
- Афонин С. А., Лунёв К. В.** Выявление тематических направлений в коллекции наборов ключевых слов 29
- Иванова К. Ф.** Приближенное решение плохо обусловленных систем линейных алгебраических уравнений с неточными коэффициентами 40

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индекс: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/pi.html E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2015

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

N. 2

February

2015

Published since September 2010

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
MIKHAILENKO B. G., Dr. Sci. (Phys.-Math.),
Acad. RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOVS JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
LIPAEV V.V., Dr. Sci. (Tech)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Korzun D. G. Service Formalism and Architectural Abstractions for Smart Space Applications	3
Bolshakov A. A., Lachugin D. V., Lobanov V. V. Design of Software of an Adaptive System of Control, Collection and Analysis of Streaming Data	13
Zharinov I. O., Zharinov O. O. The Method of Software-Based Brightness Control of Onboard Indication Equipment during Night Flight Operating Mode	23
Afonin S. A., Lunev K. V. Topic Analysis in Collection of Keyword Tuples	29
Ivanova K. F. The Approached Solutions of Ill-Conditioned SLAE with Inexact Factor	40

Information about the journal is available online at:
<http://novtex.ru/pi.html>, e-mail: prin@novtex.ru

Д. Ж. Корзун, канд. физ.-мат. наук, доц., e-mail: dkorzun@cs.karelia.ru,
Петрозаводский государственный университет

Формализм сервисов и архитектурные абстракции для программных приложений интеллектуальных пространств

Представлены результаты исследований методов построения программных приложений интеллектуальных пространств в условиях разнообразия современных вычислительных сред, повсеместных сетевых коммуникаций и наличия глобального информационного ресурса в виде сети Интернет. На основе архитектуры МЗ для интеллектуальных пространств уточняется набор основных понятий: интеллектуальное пространство, умное окружение, программное приложение, сервисы и архитектура приложения. Представлен формализм сервисов, основанный на подходе Р2Р-систем, и архитектурные абстракции для проектирования сервисов в рамках предложенного формализма.

Ключевые слова: интеллектуальные пространства, архитектура МЗ, Интернет физических устройств, сетевое взаимодействие, проектирование программных приложений

Введение

Многообразие сетевых вычислительных устройств (СВУ) меняет облик вычислительных сред для использования в повседневной деятельности человека [1–3]. Каждое СВУ управляется программным агентом, формируя "умный объект" из объекта физического мира. Наблюдаемое технологическое развитие следует известным парадигмам окружающего интеллекта (*Ambient Intelligence*), проникающих и повсеместных вычислений (*Pervasive and Ubiquitous Computing*), а также недавно появившимся парадигмам интеллектуальных пространств (*Smart Spaces*) и Интернета физических устройств (*Internet of Things*, далее — IoT).

Указанные парадигмы нацелены на обнаружение и использование ресурсов участвующих умных объектов при совместном решении задач [4–6]. Разработчики программных приложений на основе умных объектов сталкиваются с рядом трудностей. Во-первых, многообразие и различия современных аппаратно-программных платформ затрудняют прямые сетевые взаимодействия между СВУ. При проектировании аппаратной части СВУ производитель ориентируется на частные сетевые интерфейсы, что не позволяет реализовать прямое сетевое взаимодействие двух и более произвольно взятых СВУ различных производителей для всех возможных комбинаций. Многие СВУ являются мобильными, и участие умного объекта становится динамическим. Во-вторых, организация взаимодействия умных объектов, число и разнообразие которых могут

быть весьма большими, требует "интеллектуализации" — как при поиске нужных ресурсов, так и при построении требуемого в данный момент сервиса и его доставке заданному пользователю.

В данной статье обсуждается парадигма интеллектуальных пространств для разработки программных приложений, в которых построение и доставка сервисов основаны на взаимодействии умных объектов. В качестве референтного подхода используются архитектура МЗ и платформа Smart-МЗ [7], зарекомендовавшие себя в ходе создания исследовательских прототипов интеллектуальных пространств. В рассматриваемой области разработки программного обеспечения еще нет общепринятой терминологии и устоявшейся методологии. Некоторые шаги по систематизации начаты в работах [8–15]. В статье уточняются понятия интеллектуального пространства, умного окружения с разделяемым использованием знаний, программного приложения, реализуемых им сервисов и его архитектуры. В рамках проводимой систематизации выполняется формализация сервисов, согласно которой состояние вычислительной среды и ее пользователей виртуализируется в интеллектуальном пространстве, а сервисы определяются цепочками изменений в этом пространстве. Формализация сервисов позволяет ввести ряд архитектурных абстракций для использования при проектировании программных приложений. Данная работа является развитием работы [16], в том числе за счет анализа последних результатов, представленных в российской научной литературе.

IoT-среды и Семантический Веб

Развитие вычислительной и коммуникационной аппаратуры приводит к повсеместному появлению разнообразных IoT-сред, встроенных в физическое окружение человека и обеспеченных сетевой поддержкой для коммуникаций [3, 4, 15]. Такая среда может быть локальной — все СВУ сосредоточены в ограниченной физической области (комната, здание, городская площадь и т. п.). Если из среды возможен сетевой доступ к внешним ресурсам (например, Интернет-сервисам), то говорят о локализованных IoT-средах.

Разработка программных приложений для IoT-сред формирует новый объект исследований для программной инженерии. С одной стороны, IoT-технологии обеспечивают принципиальную возможность проводить повсеместные вычисления за счет связывания СВУ в локальные сети с доступом к глобальной сети Интернет [1, 17]. Устройства физического мира из ранее рассматриваемых независимо классов аппаратуры могут взаимодействовать: встроенная и бытовая техника, мобильные компьютеры и гаджеты, мультимедийное оборудование, медицинские устройства, промышленное оборудование. С другой стороны, сеть Интернет де факто стала глобальным хранилищем ресурсов. Принципиальную возможность работы с таким объемом и разнообразием ресурсов дают технологии Семантического Веба [18, 19]. В результате возникают системы, соединяющие физический и информационный миры [20]. Рассмотрим подход к построению таких систем на основе технологий IoT и Семантического Веба.

Одной из первых работ по использованию Семантического Веба для обеспечения интероперабельности (возможности взаимодействия) участников вычислительной среды была работа [21]. В этой работе была введена концепция sTuples (*Semantic Tuple Spaces*) как семантического информационного пространства для взаимодействия сервисов и их клиентов. Модель классной доски расширена за счет модели RDF представления данных (*Resource Description Framework*). Взаимодействие выполняется через обмен данными, описываемыми онтологиями. Используется модель "публикация/подписка" для построения событийно-управляемых приложений. Принципы вычислений в общем информационном пространстве сведены в обзор [22].

Многоагентный подход к построению информационного пространства IoT-среды на основе семантического брокера представлен в работах [5, 9]. В данном подходе наследуется концепция sTuples для обеспечения интероперабельности. Программные агенты (работают на подходящих СВУ) взаимодействуют друг с другом через брокера. Он поддерживает общее информационное хранилище для представления информации из глобальной сети ресурсов Семантического Веба и ее локального связывания с информацией, получаемой непосредственно из IoT-среды. По сути, предлагается декомпозиция гло-

бального Семантического Веба на множество локализованных составляющих. Дальнейшее развитие этот подход получил в архитектуре M3, реализованной в платформе Smart-M3 [7]. Аббревиатура M3 акцентирует свойства *multi-device* (множество устройств), *multi-vendor* (множество производителей аппаратуры) и *multi-domain* (множество предметных областей). Основные положения по разработке Smart-M3 приложений обсуждены в работах [8, 11]. Возникающие технологические вызовы проанализированы в работе [15].

В работах [12, 13] показано использование архитектуры M3 для локализованных IoT-сред в физических окружениях с множеством встроенных СВУ. Предлагаемые решения позволяют "открыть" информацию с окружающих физических устройств, включая маломощные, для совместного использования. Объекты физического мира становятся не просто "умными" (индивидуальное свойство), а формируют систему — интеллектуальное пространство (ИП) [6]. Совместное использование такой информации и ее семантики приводит, по сути, к возможности построения сервисов следующего поколения [8, 23, 24].

Интеллектуальные пространства

Развернутое в локализованной IoT-среде ИП позволяет организовать взаимодействие нужных СВУ для построения сервисов и доставки их пользователям. Термин "интеллектуальное" подчеркивает необходимость обеспечения построения и доставки сервисов в условиях растущих количества и разнообразия динамически участвующих СВУ, а также в условиях больших объемов вовлекаемых в обработку разнородных источников данных и учета имеющихся в них взаимосвязей.

Программная часть ИП включает две части (рис. 1): агенты и информационный связующий центр (*hub*). Участие каждого СВУ определяется программным агентом, выполняемым на этом СВУ. Пользовате-



Рис. 1. Соединение физического и информационного миров в умное окружение

ли участвуют с помощью персональных ВУ (например, смартфона) и мультимодальных интерфейсных систем (например, интерактивных экранов). Часть агентов представляют сущности информационного мира (например, Интернет-сервисы, базы данных, облачные системы).

Каждый агент может сделать часть своей информации доступной другим агентам, используя информационный центр. В свою очередь, агенту через этот центр доступна информация, поступающая от других агентов. Таким образом реализуется общее информационное пространство для взаимодействия агентов. Ключевым вопросом здесь выступает интероперабельность — возможность агентов (написанных на разных языках программирования, работающих на разных СВУ с различными программно-аппаратными платформами, в разных компьютерных сетях и т. п.) поддерживать связь и взаимодействовать друг с другом.

Будем следовать архитектуре МЗ и ее реализации в платформе Smart-МЗ. Она ориентирована на обеспечение информационной интероперабельности, в отличие от распространенных сейчас решений в виде частных сетевых протоколов, позволяющих обеспечить связь между СВУ узких классов (например, одного производителя). За счет использования технологий Семантического Веба взаимодействие сводится к обмену содержательной информацией (с точки зрения предметной области), при этом для сетевых коммуникаций используются подходящие для заданных устройств нижележащие коммуникационные IoT-технологии.

Информационный центр реализуется семантическим информационным брокером (SIB), который разворачивается на ЭВМ для заданной IoT-среды. Программные агенты называются процессорами зна-

ний (*Knowledge Processor*, далее — агент КР). Сетевое взаимодействие между агентом КР и SIB следует протоколу SSAP (*Smart Space Access Protocol*), который работает поверх различных протоколов сетевого и транспортного уровней и ориентирован на сетевые сеансы [25]. Брокер SIB поддерживает набор сетевых интерфейсов и нижележащих протоколов, а агент КР использует из них те, что поддерживаются его СВУ.

Брокер SIB использует модель RDF для представления ресурсов [19]: базовой единицей выступает утверждение-тройка "субъект—предикат—объект". Модель не зависит от предметной области, позволяя структурировать информационное содержимое ИП так, чтобы обеспечить эффективные машинную обработку и сетевой обмен. Обозначим информационное содержимое как I . Брокер SIB хранит его в RDF-хранилище. Взаимодействие агентов КР становится косвенным, сводится к публикации и чтению троек через брокера SIB, не требуются сетевые соединения непосредственно между агентами. Операция подписки позволяет агенту КР формировать постоянный запрос на отслеживание изменений некоторой части содержимого [26, 27]. Запрос сохраняется на стороне брокера SIB с обработкой при каждом изменении специфицированной части. Подписавшийся агент получает уведомление, если в этой части произошли добавления, удаления или обновления троек.

Операции агента КР над I существенно ориентированы на семантический поиск: любая операция над информационным фрагментом i требует вначале нахождения i в I . В табл. 1 сведены базовые операции агента для доступа к ИП (т. е. интерфейс прикладного разработчика, API). Реализация использует язык семантических запросов SPARQL из Семантического Веба. Результатом запроса выступает набор

Таблица 1

Базовые операции доступа к ИП: RDF-ориентированный поиск

Операция	Обозначение	Описание	Задача поиска
join, leave	Без обозначения	Сессия в виде сетевого соединения между агентом КР и брокером SIB. Агент может использовать частную онтологию o для представления информации при сетевом обмене	Определение области доступа: та часть I , к которой агент имеет права доступа
insert	$I := I + x$	Вставка новых утверждений. Набор троек x добавляется в I при предположении, что ни одна тройка из x не присутствует в I	Проверка на существование: I не содержит троек из x
remove	$I := I - x$	Удаление утверждений. Набор троек x удаляется из I при предположении, что все тройки из x присутствуют в I	Проверка на существование: I содержит x
update	$I := \langle I + x \rangle$	Обновление утверждений. Набор троек x обновляется I (атомарное выполнение операций remove и insert) при предположении, что все тройки из x уже присутствуют в I	Поиск: тройки из x для обновления в I
query	$x := [q \rightarrow I]$	Разовый поисковый запрос. Результат содержит все тройки из I , удовлетворяющие условию запроса q	Поиск: наличие в I троек, удовлетворяющих условию
subscribe, unsubscribe	await $x := [q \rightarrow I]$	Постоянный поисковый запрос. При изменении тех троек в I , которые удовлетворяют условию запроса q , формируется результат, содержащий эти изменения	Поиск: наличие изменений в I для троек, удовлетворяющих условию

троек, удовлетворяющих условию запроса q . Фактически это позволяет рассматривать брокера SIB как SPARQL-точку доступа [19]. Формулировка запроса может зависеть от выбранной онтологии.

Модель RDF позволяет использовать графовые структуры (RDF-графы) для представления информационного содержимого ИП с учетом требуемых предметных областей. Такое представление приводит к методам онтологического моделирования и соответствующим методам вывода знаний. Следуя формальной нотации из работ [15, 16], можно рассматривать ИП как (I, O) , где I — информационное содержимое, структура которого представлена в онтологии O . Разработка агента становится проблемно-ориентированной в терминах проблемной области и семантики обрабатываемых данных [29]. В то же время поддержание некой общей онтологии O для всего ИП является достаточно трудоемкой задачей. Каждый агент КР может использовать частную онтологию o (она должна быть частично согласована с теми агентами, с которыми данный агент хочет взаимодействовать). Такой подход реализован в средствах проблемно-ориентированной разработки на платформе Smart-M3 [10, 11], включая генерацию части программного кода агента по онтологии o . Для описания онтологий эти средства используют язык OWL (Web Ontology Language) из Семантического Веба.

Понятие программного приложения

Следуя работам [12, 15], определим умное окружение как состоящее из локализованной IoT-среды и развернутого в ней ИП (рис. 2). Программное приложение (ИП-приложение) является подсистемой взаимодействующих агентов КР в этом окружении, предназначенной для построения и доставки набора сервисов. Приложение использует информацию, связанную с этим окружением и его пользователями, а также выводимые из нее знания [23].

Выделяют следующие системные свойства ИП-приложения, позволяющие трактовать его сервисы как "умные" [12]. Приложение способно:

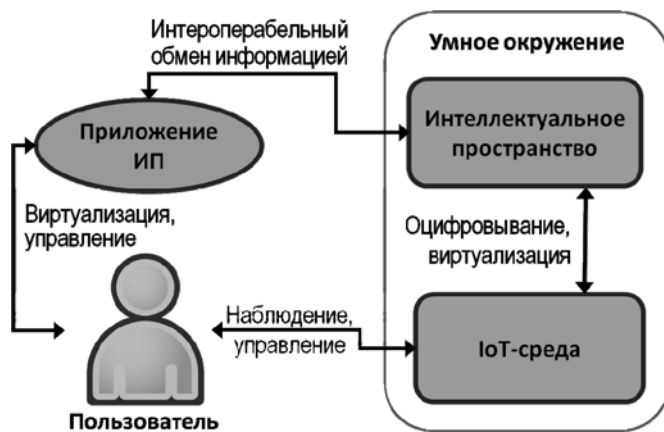


Рис. 2. Приложение ИП как сервисно-ориентированная подсистема

1) понимать текущую ситуацию, в которой оно используется, и для кого оно используется;

2) интерпретировать информационное содержимое ИП с точки зрения предметной области сервисов;

3) действовать в условиях частичной определенности (во время разработки и во время выполнения) сценариев использования.

Свойства 1 и 3 касаются использования контекста. Сервисы могут быть построены динамически и персонализировано, допускают адаптацию к текущей ситуации. Доставка сервисов может быть ситуационной и проактивной. Свойство 2 затрагивает информационную интероперабельность. Проблемно-ориентированная разработка на основе онтологий позволяет участвующим в приложении агентам КР иметь общий взгляд на окружение и его ресурсы.

Проектирование ИП-приложений основано на сценариях, включающих шаги по извлечению знаний из имеющейся информации. Сценарий определяет последовательность действий по построению сервиса, инициируемую со стороны пользователя и завершающуюся доставкой сервиса пользователю. Инициация может быть явной (команда пользователя) или неявной (обнаружение потребности на основе контекста). Доставка может быть в форме визуального представления информации пользователю (например, рекомендации) или пользователь непосредственно наблюдает изменения в физическом мире, например, освещение помещения снижается (рис. 2).

Последовательность действий сценария является событийно-управляемой: выполнить действие, если наступило событие. Такой вариант расширяется до информационно-управляемого: выполнить действие, если сформировано некоторое знание о ситуации. Причиной действия выступает появление информации в ИП и вывод знаний из этой информации. Проектирование сценария выполняется в терминах имеющейся информации и вывода знаний. Архитектурные вопросы, такие как декомпозиция приложения на агентов КР и их распределение по СВУ, решаются на следующих этапах проектирования.

В качестве иллюстрирующего примера рассмотрим следующие три программных приложения, исследовательские прототипы которых были разработаны на платформе Smart-M3 и доступны с открытым кодом.

Пример 1. Приложение M3-Weather [30] реализует погодный сервис для мобильного пользователя. Сценарий активируется изменением текущих координат местоположения пользователя (GPS-модуль телефона). Для текущего местоположения сервис показывает на экране телефона информацию о прогнозе погоды (виджет рабочего стола). В ИП связывается персональный контекст пользователя (координаты) и информация о местоположении из внешних источников (название ближайшего населенного пункта и прогноз погоды для него).

Пример 2. Приложение SmartScribo [31] реализует сервисы для мобильного мультиблогинга. В частности, позволяет пользователю работать с несколькими блогами на разных внешних блог-сервисах (например, LiveJournal, Twitter) как с одним "составным" блогом. Так, одно и то же сообщение может дублироваться в несколько блогов или из сходных сообщений (из нескольких блогов) формируется одно.

Пример 3. Приложение SmartRoom [32] реализует сервисы интеллектуального зала для поддержки таких мероприятий коллаборативной деятельности, как конференции и семинары. Во время мероприятия в помещении зала один проекционный экран используется для презентаций докладчиков, второй — для отображения текущей программы конференции. Программа конференции адаптируется под текущую ситуацию: реально затраченное время доклада, отмена докладов, добавление нового и т. п. Управление презентацией ведется докладчиками со своих мобильных устройств (смартфон, нетбук). Остальные участники могут получать информацию о ходе мероприятия и влиять на процесс также со своих мобильных устройств.

Формализм сервисов

На основе представленного сценарного подхода будем концептуально рассматривать сервис как процедуру извлечения знаний из информационного содержимого I с формированием на их основе информации для доставки пользователям. Шаги этой процедуры приведены на рис. 3. На этом уровне описания сервиса не определяется, какие конкретно агенты КР и как реализуют эти шаги.

Информационный сервис предназначен для предоставления пользователю полезной информации. На шаге 1 ожидается наступление события, активирующего построение сервиса (пользователю нужна информация). На шаге 2 формируется нужная пользователю информация x . На шаге 3 определяется, какие устройства d отвечают за доставку x пользова-

телю. На шаге 4 происходит обновление визуального содержимого v_d на устройстве d , чтобы отобразить x .

Управляющий сервис позволяет влиять на работу сущностей физического или информационного миров. На шаге 1 ожидается наступление события для активации управления. На шагах 2 и 3 на основе анализа имеющегося в ИП информационного содержимого определяется, какие изменения у сервиса должен внести. После шага 4 целевой сущности становится доступна информация с описанием требуемых от нее действий.

Онтологическое представление позволяет трактовать I как набор фактических информационных объектов и семантических связей между ними. Структура этих объектов и связей определяется онтологиями (например, OWL-классы, отношения и ограничения). Каждый такой объект может быть представлен как экземпляр онтологического класса (например, OWL-индивид) с собственными значениями свойств данных и объектных свойств. Объектные свойства позволяют представить в I семантические связи между объектами. Это соответствует известному понятию OWL-графа индивидов.

Такая трактовка информационного содержимого ИП может быть обобщена на основе P2P-модели, представленной в работе [28]. Модель описывает виртуализацию в ИП объектов физического и информационных миров, самих агентов КР и объектов для представления выводных информации и знаний. Поверх I формируется P2P-сеть G_I , где роль узлов (*peer*) выполняют объекты. Внесение агентами изменений в I (insert, update, remove) изменяет сеть G_I , как это и происходит в P2P-сетях в силу динамической ротации узлов и выбора соседей [33]. Тогда информационное содержимое можно рассматривать как систему взаимодействующих объектов, которые, с одной стороны, выступают активными сущностями (выполняют действия) и, с другой стороны, сами подвергаются изменениям (следствие действий).

В результате сервис ИП-приложения можно формализовать как пошаговый процесс (поток) изме-

Информационный сервис	
Требуется: Онтология o описывает структуру информационного содержимого ИП.	
Множество U определяет множество доступных интерфейсных СВУ	
Шаг 1: await $[q_{\text{act}}(o) \rightarrow I] = \text{true}$	{ активация по событию }
Шаг 2: query $x := [q_{\text{info}}(o) \rightarrow I]$	{ извлечение информации, вывод знаний }
Шаг 3: decide d in U	{ выбор интерфейсного СВУ }
Шаг 4: visualize $v_d := v_d + x$	{ доставка сервиса }
Управляющий сервис	
Требуется: Онтология o описывает структуру информационного содержимого ИП	
Шаг 1: await $[q_{\text{act}}(o) \rightarrow I] = \text{true}$	{ активация по событию }
Шаг 2: query $x := [q_{\text{info}}(o) \rightarrow I]$	{ извлечение информации, вывод знаний }
Шаг 3: decide $y := f(x, o)$	{ формирование управляющего действия }
Шаг 4: update $I := \langle I + y \rangle$	{ доставка сервиса }

Рис. 3. Шаги построения сервисов (адаптированы из работы [28])

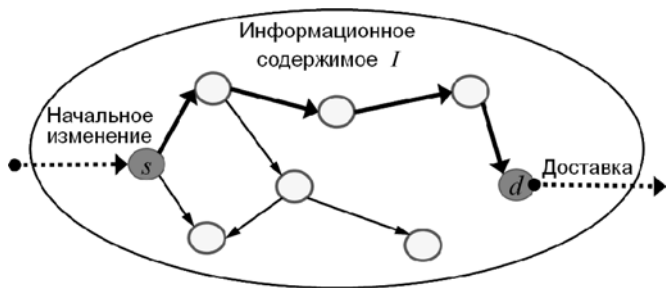


Рис. 4. Построение сервиса как P2P-пути $s \rightarrow^* d$

нений объектов в I . Изменение объекта $i_1 \in I$ формирует событие, наблюдаемое другими объектами. Они выполняют действия, приводящие к изменению объекта i_2 (или его появлению в I). Такой процесс продолжается далее. Допускается ветвление, когда изменение одного объекта влияет на действия нескольких объектов.

Пусть для сервиса задан начальный объект $s \in I$. Рассмотрим маршрутный граф $D(s)$, получаемый как все пути потока изменений в G_I , начинающегося при изменении s . Построение сервиса соответствует пути $s \rightarrow^* d$, как схематично показано на рис. 4. Выбор финального объекта d реализуется заданным агентом КР, который считывает из I информацию об этом объекте и доставляет сервис пользователю.

Представленная формализация весьма гибкая для описания возможных сервисов. Это достигается за счет выбора параметров: начальный и финальный объекты, путь $s \rightarrow^* d$ в графе $D(s)$. Отметим, что такой подход ранее показал свою применимость для решения задач маршрутизации в P2P-сетях [33, 34]. Рассмотрим примеры на основе приведенных в предыдущем разделе приложений.

Пример 1 (приложение M3-Weather). Информационный сервис основан на извлечении погодной информации по текущим координатам. В ИП представляются объекты для:

- координат (начальный объект, значения снимаются с мобильного телефона);
- населенного пункта (название получается по координатам с помощью внешнего веб-сервиса);
- погоды (финальный объект, прогноз получается по названию населенного пункта с помощью внешнего веб-сервиса).

При изменении координат выполняется последовательное обновление значений для населенного пункта и прогноза погоды.

Пример 2 (приложение SmartScribo). Сообщения из блогов выступают объектами для виртуализации в ИП. Сервисы являются информационными и реализуют поиск нужных сообщений для пользователя в месте их основного хранения — блог-сервисы в сети Интернет. Если сервис позволяет вносить изменения на блог-сервис, то такой сервис становится управляющим. Сообщения связаны с пользователями в первую очередь отношением "автор". Простейшим вариантом, когда изменение одного сообще-

ния может вызвать изменения в других, является дублирование сообщения в несколько блогов. Более сложный процесс возникает, когда измененное (или появившееся новое) сообщение удовлетворяет условию "сходность" с другими сообщениями. Тогда в ИП создается агрегированное сообщение, которое существует только в ИП, объединяя несколько реальных сообщений. Возможно также связывание в ИП сообщений с пользователями и отношением "интерес", когда пользователю u интересна тематика сообщения m .

Пример 3 (приложение SmartRoom). Объектами для виртуализации в ИП выступают участники мероприятия и их материалы (презентации). Текущий слайд презентации является изменяемым объектом во время доклада. Эти изменения должны быть доставлены на проекционный экран в помещении, а также участникам (если последние наблюдают за докладом со своего мобильного устройства). Такую простую связь "слайд—отображение" можно расширять за счет привязки слайдов к медиа-ресурсам (например, докладчик для данного слайда показывает видеоматериал) или за счет динамического построения слайда (например, проекционный экран является интерактивной доской с возможностью построения дополнительных изображений на слайде).

Архитектурные абстракции

Рассмотрим процесс построения сервиса согласно пути $s \rightarrow^* d$. В последнем представлены два типа переходов: онтологические и посреднические. Онтологический переход $i \rightarrow j$ определяется объектными свойствами из онтологии, связывающими экземпляры классов (OWL-индивиды) в I . Эта связь хранится непосредственно в I . При построении сервиса она проявляется в результате выполнения поисковых запросов: от объекта i переходим к анализу онтологически связанного с ним объекта j . Таким образом, этот переход при построении сервиса происходит на стороне брокера SIB, не затрагивая архитектуру ИП-приложения.

Посреднический переход $i \rightarrow j$ является результатом действий некоторого агента КР. Последний анализирует i и на основании имеющихся у агента знаний выполняет изменение объекта j . Таким образом, посреднические переходы являются определяющими для проектирования архитектуры ИП-приложения как системы агентов КР. Необходимо построить разбиение посреднических переходов на непересекающиеся группы, каждая группа соответствует агенту КР, отвечающему за построение этих переходов.

Взаимодействие каждого агента КР в системе определяется его индивидуальной логикой в рамках схемы (i, q, J) . Исходный объект $i \in I$ определяет событие (изменение i), активирующее агента, а сам объект содержит информацию для действий агента. Запрос q позволяет реализовать онтологические переходы от i при построении сервиса: поиск в I дополнительной информации, семантически связанной с

Таблица 2

Архитектурные абстракции для ИП-приложения

Шаблон	Описание
P-C	Производитель—Потребитель. Агент p публикует объект в I . Агент c находит этот объект и реагирует
Pipe	Набор агентов KP_0, KP_1, \dots, KP_n формируют аналог цепи потребления (линейная) с источником KP_0 и получателем KP_n . Шаблон P-C является частным случаем при $n = 1$
Tree	Иерархическое ветвление, когда агент KP вносит изменения, вызывающие реакцию нескольких агентов KP. Реализуется синхронизация "один-к-многим" с эпидемическим распространением изменений в I
Flow	Обобщение шаблона Tree. Допускаются циклические цепочки при взаимодействии агентов

исходным объектом. На основании всей полученной из ИП информации и объединяя ее с собственными знаниями, агент формирует множество объектов J для их обновления или создания в I . В результате формируются пути $i \rightarrow^* j, j \in J$.

Индивидуальная логика агентов приводит к взаимодействию параллельно работающих агентов KP при построении сервиса. В табл. 2 представлены ар-

хитектурные абстракции, описывающие шаблоны взаимодействия.

При проектировании эти абстракции конкретизируются на основе таксономии агентов KP, предложенной в работе [12]. Определяется функциональный тип агента KP в процессе построения сервиса (табл. 3). Каждому агенту KP назначается нужный тип (или их комбинация), определяющий дальнейшую разработку индивидуальной логики агента.

Пример 1 (приложение M3-Weather). Возникает цепь потребления: KP_0 — это клиент на мобильном телефоне (публикует координаты и затем отслеживает появление прогноза погоды); KP_1 — это контроллер для обновления названия населенного пункта при изменении координат; KP_2 — это контроллер для обновления прогноза погоды при изменении населенного пункта. Также KP_1 и KP_2 выступают фильтрами для соответствующих внешних веб-сервисов.

Пример 2 (приложение SmartScribo). Взаимодействие объектов-сообщений является иерархическим, поскольку решается задача поиска сообщений (среди большого множества) для предоставления пользователю. Помимо клиентских агентов KP, в архитектуру входят агенты, реализующие адаптеры и фильтры для блог-сервисов. Также используются агрегаторы, в задачи которых входит поиск сходных сообщений и выработка рекомендаций.

Таблица 3

Таксономия агентов KP по функциональному типу

Тип	Описание агента KP	Задача	Пример
Адаптер (<i>Adapter</i>)	Представляет внешнюю сущность физического или информационного миров, подключаемую к ИП	Обеспечивать ИП информацией от внешней сущности и наоборот, реализуя информационную виртуализацию этой сущности в ИП	Сенсор, предмет бытовой электроники, Интернет-ресурс
Клиент (<i>Client</i>)	Представляет пользователя-человека	Обеспечивать ИП информацией о пользователе и доставлять сервисы пользователю	Персональные мобильные компьютеры, такие как смартфоны и нетбуки
Агрегатор (<i>Aggregator, Reasoner</i>)	Выполняет анализ накопленного информационного содержимого	Извлекать знания из имеющейся информации, включая публикацию результатов в ИП	Построение прогнозов и рекомендаций
Контроллер (<i>Controller</i>)	Выполняет динамическое обновление семантических связей в ИП	Обеспечивать отражение в ИП текущего состояния IoT-окружения и контекста	Отслеживание возможности использования ресурса среды для построения персонализированного сервиса
Фильтр (<i>Filter</i>)	Выполняет отбор и, при необходимости, преобразование информации из внешних источников для публикации в ИП	Обеспечивать ИП содержимым на основе отбора информации из одного или более источников физического или информационного миров	Формирование сводки о публикационной активности заданного пользователя из системы Google Scholar
Монитор (<i>Monitor</i>)	Отслеживает заданные изменения в ИП и ведет соответствующую историю	Извлекать информацию о проходящих в ИП процессах во времени	Запись последовательности входа и выхода СВУ из ИП

Пример 3 (приложение SmartRoom). Взаимодействие объектов может быть циклическим. Так, показ слайда может вызвать появление новой информации как от самого докладчика (например, изменение слайда во время доклада), так и от остальных участников (например, генерация вопросов по слайду и связь с материалами других докладчиков). Интересной является возможность использования мониторов для отслеживания хода процесса и построения сводки (в виде слайдов) по завершении мероприятия.

Предложенные формализация сервисов и архитектурные абстракции определяют следующие задачи проектирования сервисов ИП-приложения.

1. Разработка концепции сервисов в терминах извлечения знаний и формирования на их основе информации для доставки сервиса.

2. Описание информационного содержимого ИП как системы связанных объектов, виртуализирующей все участвующие сущности физического и информационного миров.

3. Анализ путей построения сервисов в системе связанных объектов для определения набора агентов КР, необходимого для реализации сервиса.

4. Построение архитектуры приложения на основе архитектурных абстракций и определение функционального типа каждого агента КР.

5. Разработка индивидуальной логики каждого агента КР на основе его функционального типа и реализуемых посреднических переходов при построении сервисов.

Заключение

В работе описана проблема проектирования программных приложений интеллектуального пространства, реализующих построение и доставку сервисов в условиях современных вычислительных сред. С одной стороны, в рассматриваемой области еще нет достаточной и устоявшейся терминологии. В статье на основе ряда недавних публикаций в международной и российской научной литературе уточняются основные понятия интеллектуальных пространств как парадигмы разработки программного обеспечения для Интернета физических устройств. С другой стороны, интеллектуальные пространства ориентированы на слабосвязные многоагентные приложения, что требует новых методов по сравнению с традиционными распределенными системами. В статье представлен формализм сервисов и предложены архитектурные абстракции для использования при проектировании подобных приложений.

Работа выполнена при финансовой поддержке Минобрнауки России по заданию № 2014/154 на выполнение государственных работ в сфере научной деятельности в рамках базовой части государственного задания, НИР № 1481. Статья подготовлена при поддержке федеральной целевой программы "Исследования и разработки по приоритетным направлениям

развития научно-технологического комплекса России на 2014–2020 годы", соглашение № 14.574.21.0060 (RFMEFI57414X0060).

Список литературы

1. Kortuem G., Kawsar F., Sundramoorthy V., Fitton D. Smart objects as building blocks for the internet of things // IEEE Internet Computing. 2010. Vol. 14, Is. 1. P. 44–51.
2. Cook D. J., Das S. K. Pervasive computing at scale: Transforming the state of the art // Pervasive and Mobile Computing. 2012. Vol. 8, Is. 1. P. 22–35.
3. Гаврилов А. В. Искусственный домовой // Искусственный интеллект и принятие решений. 2012. № 2. С. 77–89.
4. Cook D. J., Das S. K. How smart are our environments? An updated look at the state of the art // Pervasive and Mobile Computing. 2007. Vol. 3, Is. 2. P. 53–73.
5. Oliver I., Boldyrev S. Operations on Spaces of Information web // Proc. of 2009 IEEE 3rd Int. Conf. on Semantic Computing (ICSC). 14–16 Sept. 2009. Berkeley, CA, USA. IEEE, 2009. P. 267–274. URL: <http://dx.doi.org/10.1109/ICSC.2009.102>
6. Юсупов Р. М., Ронжин А. Л. От умных приборов к интеллектуальному пространству // Вестник РАН. 2010. Т. 80, № 1. С. 45–51.
7. Honkola J., Laine H., Brown R., Tyrkko O. Smart-M3 information sharing platform // Proc. IEEE Symp. Computers and Communications (ISCC), 22–25 June 2010, Riccione, Italy. Washington: IEEE Computer Society, 2010. P. 1041–1046. URL: <http://dx.doi.org/10.1109/ISCC.2010.5546642>
8. Balandin S., Waris H. Key properties in the development of smart spaces // Proc. of 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI), 19–24 July 2009. San Diego, CA, USA. Part II: Intelligent and Ubiquitous Interaction Environments. LNCS 5615. Springer Berlin Heidelberg, 2009. P. 3–12. URL: http://dx.doi.org/10.1007/978-3-642-02710-9_1
9. Smirnov A., Kashnevik A., Shilov N. etc. Anonymous agent coordination in smart spaces: State-of-the-art // Proc. of 9th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN) and 2nd Conf. Smart Spaces (ruSMART), 15–18 Sept. 2009. St.-Petersburg, Russia. LNCS 5764. Springer Berlin Heidelberg, 2009. P. 42–51. URL: http://dx.doi.org/10.1007/978-3-642-04190-7_5
10. Palviainen M., Katsanos A. Model and ontology-based development of smart space applications // Pervasive Computing and Communications Design and Deployment: Technologies, Trends, and Applications. IGI Global, 2011. P. 126–148. URL: <http://dx.doi.org/10.4018/978-1-60960-611-4>
11. Корзун Д. Ж., Ломов А. А., Ваняг П. И. Автоматизированная модельно-ориентированная разработка программных агентов для интеллектуальных пространств на платформе Smart-M3 // Программная инженерия. 2012. № 5. С. 6–14.
12. Ovaska E., Cinotti T. S., Toninelli A. The design principles and practices of interoperable smart spaces // Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools. IGI Global, 2012. P. 18–47. URL: <http://dx.doi.org/10.4018/978-1-60960-735-7.ch002>
13. Kiljander J., Ylisaukko-oja A., Takalo-Mattila J. etc. Enabling semantic technology empowered smart spaces // Journal of Computer Networks and Communications. 2012. P. 1–14.
14. Bernini D., Fiamberti F., Micucci D., Tisato F. Architectural abstractions for spaces-based communication in smart environments // Journal of Ambient Intelligence and Smart Environments. 2012. Vol. 4, Is. 3. P. 253–277.
15. Korzun D., Balandin S., Gurtov A. Deployment of Smart Spaces in Internet of Things: Overview of the design challenges // Proc. of 13th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN) and 6th Conf. on Internet of Things and Smart Spaces (ruSMART). 28–30 Aug. 2013. St.-Petersburg, Rus-

sia. LNCS 8121. Springer Berlin Heidelberg, 2013. P. 48–59. URL: http://dx.doi.org/10.1007/978-3-642-40316-3_5

16. **Korzun D.** Service Formalism and Architectural Abstractions for Smart Space Applications // Proc. of Central & Eastern European Software Engineering Conference in Russia (CEE-SEC(R)). 23–25 Oct. 2014. Moscow, Russia. ACM, 2014. URL: <http://dx.doi.org/10.1145/2687233.2687253>

17. **Atzori L., Iera A., Morabito G.** The internet of things: A survey // Computer Networks. 2010. Vol. 54, Is. 15. P. 2787–2805.

18. **Хорошевский В. Ф.** Пространства знаний в сети интернет и semantic web (часть I) // Искусственный интеллект и принятие решений. 2008. № 1. С. 80–97.

19. **Gutierrez C., Hurtado C. A., Mendelzon A. O., Perez J.** Foundations of semantic web databases // Journal of Computer and System Sciences. 2011. Vol. 77, Is. 3. P. 520–541.

20. **Соколов Б. В., Юсупов Р. М.** Неокибернетика в структуре современных знаний // Робототехника и техническая кибернетика. 2014. № 2 (3). С. 3–11.

21. **Khushraj D., Lassila O., Finin T. W.** sTuples: Semantic tuple spaces // Proc. 1st Annual Int'l Conf. Mobile and Ubiquitous Systems (MobiQuitous). 22–26 Aug. 2004, Boston, USA. IEEE, 2004. P. 268–277. URL: <http://dx.doi.org/10.1109/MOBIQ.2004.1331733>

22. **Nixon L. J. B., Simperl E., Krummenacher R., Martinrecuerda F.** Tuplespace-based computing for the semantic web: A survey of the state-of-the-art // Knowledge Engineering Review. 2008. Vol. 23, Is. 2. P. 181–212.

23. **Городецкий В. И.** Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации // Известия РАН. "Теория и системы управления". 2012. № 2. С. 92–120.

24. **Смирнов А. В., Левашова Т. В.** Принципы и модели контекстно-управляемой интеграции знаний // Информационные технологии и вычислительные системы. 2013. № 4. С. 58–73.

25. **Ломов А. А.** Взаимодействие программного агента на уровне сессии с интеллектуальным пространством // Ученые записки Петрозаводского государственного университета. 2013. Вып. 8 (137). С. 118–122.

26. **Ломов А. А., Корзун Д. Ж.** Операция подписки для приложений в интеллектуальных пространствах платформы Smart-M3 // Труды СПИИРАН. 2012. Вып. 4 (23). С. 439–458.

27. **Галов И. В., Корзун Д. Ж.** Модель уведомлений для разработки программных приложений интеллектуальных пространств // Труды СПИИРАН. 2014. Вып. 35. С. 189–211.

28. **Korzun D., Balandin S.** A peer-to-peer model for virtualization and knowledge sharing in smart spaces // Proc. of 8th Int'l Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM). 24–28 Aug. 2014. Rome, Italy. IARIA XPS Press, 2014. P. 87–92. URL: http://www.thinkmind.org/index.php?view=article&articleid=ubicomm_2014_3_50_10195

29. **Разумовский А. Г., Пантелеев М. Г.** Валидация объектно-ориентированных программ с использованием онтологии // Программная инженерия. 2012. № 7. С. 7–13.

30. **Samoryadova A., Galov I., Borovinskiy P. etc.** M3-weather: A Smart-M3 world-weather application for mobile users // Proc. of 8th Conf. of Open Innovations Framework Program FRUCT, 9–12 Nov. 2010, Lappeenranta, Finland. SPb: SUAI, 2010. P. 160–166. URL: <http://www.fruct.org/publications/fruct8/files/Sam.pdf>

31. **Korzun D., Galov I., Balandin S.** Proactive personalized mobile multiblogging service on SmartM3 // Journal of Computing and Information Technology. 2012. Vol. 20, N. 3. P. 175–182.

32. **Korzun D., Galov I., Kashevnik A., Balandin S.** Virtual shared workspace for smart spaces and M3-based case study // Proc. of 15th Conf. of Open Innovations Association FRUCT. 21–25 Apr. 2014. St.-Petersburg, Russia. SPb.: ITMO, 2014. P. 60–68. URL: <http://www.fruct.org/publications/fruct15/files/Kor.pdf>

33. **Korzun D., Gurtov A.** Structured Peer-to-Peer Systems: Fundamentals of Hierarchical Organization, Routing, Scaling, and Security. Springer, 2013. 366 p.

34. **Корзун Д. Ж., Гуртов А. В.** Использование линейных диофантовых уравнений для моделирования маршрутизации в самоорганизующихся сетях // Электросвязь. 2006. № 6. С. 34–38.

D. G. Korzun, Associate Professor, e-mail: dkorzun@cs.karelia.ru, Petrozavodsk State University

Service Formalism and Architectural Abstractions for Smart Space Applications

The paper presents results of research on creating software applications of smart spaces when computing environments are diverse in devices, network communications are ubiquitous, and the Internet can be used as a global information resource. The reference case is the M3 architecture, based on which we sequentially elaborate the notions of smart space, smart environment of knowledge sharing, smart space application, its services and its architecture. We consider P2P-based service formalism, on the top of which several architectural abstractions are presented. They provide means for designing smart space applications.

Keywords: smart spaces, M3 architecture, Internet of Things, network-based interaction, software application, smart service, software architecture

References

1. **Kortuem G., Kawsar F., Sundramoorthy V., Fitton D.** Smart objects as building blocks for the internet of things. *IEEE Internet Computing*. 2010. Vol. 14, Is. 1. P. 44–51.

2. **Cook D. J., Das S. K.** Pervasive computing at scale: Transforming the state of the art. *Pervasive and Mobile Computing*. 2012. Vol. 8, Is. 1. P. 22–35.

3. **Gavrilo A. V.** Iskusstvennyj domovoj. *Iskusstvennyj intellekt i prinyatie reshenij*. 2012. N 2. P. 77–89 (in Russian).

4. **Cook D. J., Das S. K.** How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*. 2007. Vol. 3, Is. 2. P. 53–73.
5. **Oliver I., Boldyrev S.** Operations on Spaces of Information web. *Proc. of 2009 IEEE 3rd Int. Conf. on Semantic Computing (ICSC)*. 14–16 Sept. 2009. Berkeley, CA, USA. P. 267–274. URL: <http://dx.doi.org/10.1109/ICSC.2009.102>
6. **Yusupov R. M., Ronzhin A. L.** Ot umnyx priborov k intellektual'nomu prostranstvu. *Vestnik RAN*. 2010, Vol. 80, N. 1. P. 45–51 (in Russian).
7. **Honkola J., Laine H., Brown R., Tyrkko O.** Smart-M3 information sharing platform. *Proc. of IEEE Symp. Computers and Communications (ISCC)*. 22–25 June 2010, Riccione, Italy. Washington: IEEE Computer Society, 2010. P. 1041–1046. URL: <http://dx.doi.org/10.1109/ISCC.2010.5546642>
8. **Balandin S., Waris H.** Key properties in the development of smart spaces. *Proc. of 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI)*. 19–24 July 2009. San Diego, CA, USA. Part II: Intelligent and Ubiquitous Interaction Environments. LNCS 5615. Springer Berlin Heidelberg, 2009. P. 3–12. URL: http://dx.doi.org/10.1007/978-3-642-02710-9_1
9. **Smirnov A., Kashnevik A., Shilov N., Oliver I., Balandin S., Boldyrev S.** Anonymous agent coordination in smart spaces: *State-of-the-art*. *Proc. of 9th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN) and 2nd Conf. Smart Spaces (ruSMART)*. 15–18 Sept. 2009. St.-Petersburg, Russia. LNCS 5764. Springer Berlin Heidelberg, 2009. P. 42–51. URL: http://dx.doi.org/10.1007/978-3-642-04190-7_5
10. **Palviainen M., Katasonov A.** Model and ontology-based development of smart space applications. *Pervasive Computing and Communications Design and Deployment: Technologies, Trends, and Applications*. IGI Global, 2011. P. 126–148. URL: <http://dx.doi.org/10.4018/978-1-60960-611-4>
11. **Korzun D., Lomov A., Vanag P.** Avtomatizirovannaya model'no-orientirovannaya razrabotka programmyx agentov dlya intellektual'nyx prostranstv na platforme Smart-M3. *Programmnyaya ingeneriya*. 2012. N. 5. P. 6–14 (in Russian).
12. **Ovaska E., Cinotti T. S., Toninelli A.** The design principles and practices of interoperable smart spaces. In Book: *Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools*. IGI Global, 2012. P. 18–47. URL: <http://dx.doi.org/10.4018/978-1-60960-735-7.ch002>
13. **Kiljander J., Ylisaukko-oja A., Takalo-Mattila J., Etelape-ra M., Soininen J.-P.** Enabling semantic technology empowered smart spaces. *Journal of Computer Networks and Communications*. 2012. P. 1–14.
14. **Bernini D., Fiamberti F., Micucci D., Tisato F.** Architectural abstractions for spaces-based communication in smart environments. *Journal of Ambient Intelligence and Smart Environments*. 2012. Vol. 4, Is. 3. P. 253–277.
15. **Korzun D., Balandin S., Gurtov A.** Deployment of Smart Spaces in Internet of Things: Overview of the design challenges. *Proc. of 13th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN) and 6th Conf. on Internet of Things and Smart Spaces (ruSMART)*. 28–30 Aug. 2013. St.-Petersburg, Russia. LNCS 8121. Springer Berlin Heidelberg, 2013. P. 48–59. URL: http://dx.doi.org/10.1007/978-3-642-40316-3_5
16. **Korzun D.** Service Formalism and Architectural Abstractions for Smart Space Applications. *Proc. of Central & Eastern European Software Engineering Conference in Russia (CEE-SEC(R))*. 23–25 Oct. 2014. Moscow, Russia. ACM, 2014. URL: <http://dx.doi.org/10.1145/2687233.2687253>
17. **Atzori L., Iera A., Morabito G.** The internet of things: A survey. *Computer Networks*. 2010. Vol. 54, Is. 15. P. 2787–2805.
18. **Xoroshevskij V. F.** Prostranstva znaniy v seti internet i semantic web (chast' 1). *Iskustvennyy intellekt i prinyatie reshenij*. 2008. N 1. P. 80–97 (in Russian).
19. **Gutierrez C., Hurtado C. A., Mendelzon A. O., Perez J.** Foundations of semantic web databases. *Journal of Computer and System Sciences*. 2011. Vol. 77, Is. 3. P. 520–541.
20. **Sokolov B. V., Yusupov R. M.** Neokibernetika v strukture sovremennyx znaniy. *Robototexnika i texnicheskaya kibernetika*. 2014. N 2 (3). P. 3–11 (in Russian).
21. **Khushraj D., Lassila O., Finin T. W.** sTuples: Semantic tuple spaces. *Proc. of 1st Annual Int'l Conf. Mobile and Ubiquitous Systems (MobiQuitous) 22–26 Aug. 2004*. Boston, USA. IEEE, 2004. P. 268–277. URL: <http://dx.doi.org/10.1109/MOBIQ.2004.1331733>
22. **Nixon L. J. B., Simperl E., Krummenacher R., Martin-recuerda F.** Tuplespace-based computing for the semantic web: A survey of the state-of-the-art. *Knowledge Engineering Review*. 2008. Vol. 23, Is. 2. P. 181–212.
23. **Gorodetsky V.** Samoorganizaciya i mnogoagentnye sistemy. I. Modeli mnogoagentnoj samoorganizacii. *Izvestiya RAN. "Teoriya i sistemy upravleniya"*. 2012. N. 2. P. 92–120 (in Russian).
24. **Smirnov A. V., Levashova T. V.** Principy i modeli kontekstno-upravlyаемoj integracii znaniy. *Informacionnye texnologii i vychislitel'nye sistemy*. 2013. N. 4. P. 58–73 (in Russian).
25. **Lomov A. A.** Vzaimodejstvie programmnogo agenta na urovne sessii s intellektual'nyx prostranstvom. *Uchenye zapiski Petrozavodskogo gosudarstvennogo universiteta*. 2013. Is.8 (137). P. 118–122 (in Russian).
26. **Lomov A. A., Korzun D.** Operaciya podpiski dlya prilozhenij v intellektual'nyx prostranstvax platformy Smart-M3. *Trudy SPI-IRAN – SPIIRAS Proceedings*. 2012. Is. 4 (23). P. 439–458 (in Russian).
27. **Galov I. V., Korzun D.** Model' uvedomlenij dlya razrabotki programmyx prilozhenij intellektual'nyx prostranstv. *Trudy SPI-IRAN – SPIIRAS Proceedings*. 2014. Is. 35. P. 189–211 (in Russian).
28. **Korzun D., Balandin S.** A peer-to-peer model for virtualization and knowledge sharing in smart spaces. *Proc. of 8th Int'l Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*. 24–28 Aug. 2014. Rome, Italy. IARIA XPS Press, 2014. P. 87–92. URL: http://www.thinkmind.org/index.php?view=article&articleid=ubicomm_2014_3_50_10195
29. **Razumovskij A. G., Panteleev M. G.** Validaciya ob'ektno-orientirovannyx programms s ispol'zovaniem ontologii. *Programmnyaya ingeneriya*. 2012. N 7. Pp. 7–13 (in Russian).
30. **Samoryadova A., Galov I., Borovinskiy P., Kulakov K., Korzun D.** M3-weather: A Smart-M3 world-weather application for mobile users. *Proc. of 8th Conf. of Open Innovations Framework Program FRUCT*. 9–12 Nov. 2010. Lappeenranta, Finland. SPb: SUAI, 2010. P. 160–166. URL: <http://www.fruct.org/publications/fruct8/files/Sam.pdf>
31. **Korzun D., Galov I., Balandin S.** Proactive personalized mobile mutliblogging service on SmartM3. *Journal of Computing and Information Technology*, 2012. Vol. 20, N. 3. P. 175–182.
32. **Korzun D., Galov I., Kashevnik A., Balandin S.** Virtual shared workspace for smart spaces and M3-based case study. *Proc. of 15th Conf. of Open Innovations Association FRUCT*. 21–25 Apr. 2014. St.-Petersburg, Russia. SPb.: ITMO, 2014. P. 60–68. URL: <http://www.fruct.org/publications/fruct15/files/Kor.pdf>
33. **Korzun D., Gurtov A.** *Structured Peer-to-Peer Systems: Fundamentals of Hierarchical Organization, Routing, Scaling, and Security*. Springer, 2013. 366 p.
34. **Korzun D., Gurtov A.** Ispol'zovanie linejnyx Diofantovyx uravnenij dlya modelirovaniya marshrutizacii v samoorganizuyushixsya setyax. *E'lektrosvyaz'*. 2006. N. 6. P. 34–38 (in Russian).

А. А. Большаков, д-р техн. наук, проф., e-mail: aabolshakov57@gmail.com, Санкт-Петербургский государственный политехнический университет,
Д. В. Лачугин, аспирант, ассистент, e-mail: lachugindm@mail.ru,
В. В. Лобанов, канд. техн. наук, докторант, e-mail: lobanovv@gmail.com, Саратовский государственный технический университет имени Гагарина Ю. А.

Проектирование программного комплекса адаптивной системы управления, сбора и анализа потоковых данных

Предложена схема адаптивной информационно-измерительной системы управления, сбора и анализа потоковой сигнальной информации. Ее отличительной особенностью является способность динамически изменять функциональную конфигурацию, чтобы сократить общее время цикла обработки системой потоковой информации. Также рассматривается синтез программного комплекса серверной и клиентской частей предложенной реконфигурируемой вычислительной системы.

Ключевые слова: адаптивная система управления, потоковая информация, реконфигурируемая вычислительная система, программный комплекс

Введение

К настоящему времени возросла сложность и объем решаемых вычислительных задач, что обусловило необходимость создания распределенных комплексов. В связи с этим увеличилась потребность информационного обмена при функционировании многих прикладных систем, которые осуществляют передачу, анализ и хранение данных, получаемых от объектов наблюдения преобразователями различного назначения. Такие преобразователи являются чувствительными элементами измерительного, регулирующего, сигнального или управляющего характера. Их основной задачей является преобразование выходной величины объекта управления в сигнальную информацию, удобную для измерения, передачи, преобразования и хранения, т. е. трансформирование в более высокоуровневый поток данных, с формированием канала потоковой сигнальной информации. Таким образом, большинство систем передачи данных можно отнести к потоковым, так как они обеспечивают "доставку информации" потребителям в виде потоков. При таком виде передачи информации можно начинать работать с данными при их получении, не дожидаясь полной загрузки. Потоквым методом можно передавать аудио-, видеоконтент, а также данные, структурированные в произвольном формате. Особый интерес представляют видеопотоки, они являются основой функционирования многих распределенных систем

мониторинга, регистрации событий, ориентации и управления движением объектами с видеопотоками в качестве обратной связи и т. д. Ядром большинства современных прикладных систем являются коммуникационные протоколы, которые осуществляют координацию процессов передачи массивов данных. При этом строятся распределенные системы сбора и обработки информации, узлы которых соединяются с использованием сети Интернет.

Единая информационная среда, в которой функционируют современные прикладные системы потокового вещания (как правило, видео- и звукового контента), требует учета ограничений, накладываемых сетью Интернет. Одно из основных направлений развития и повышения производительности систем заключается в эффективном использовании пропускной способности каналов передачи. При этом возрастает интерес к применению адаптивных систем кодирования и передачи потоковых данных. Такие системы основываются на современных протоколах прикладного уровня, в основе которых содержатся классические HTTP (HTTP Adaptive Streaming, Smooth Streaming, Adobe HTTP Dynamic Streaming) и RTSP (3GPP PSS) транспорты [1]. При этом адаптация качества предполагает переход между "готовыми" потоками с различными битрейтами. Преимуществом этой технологии является достаточно высокая надежность и простота в эксплуатации. Однако требуется большое число потоков, чтобы обеспечить различные комбинации разрешения и качества, что,

в свою очередь, требует более производительных, специализированных и дорогостоящих приемопередатчиков. Поэтому применение этих технологий неприемлемо в конечных узлах распределенных информационных систем.

Кроме этого, к современным распределенным информационным системам обычно предъявляют различные требования, такие как работа в режиме реального масштаба времени, оперирование значительными объемами принимаемой информации. Интерес многих исследователей направлен на изучение возможностей увеличения производительности распределенных вычислительных комплексов для удовлетворения описанных выше условий. Для этого в настоящее время активно применяют возможности распараллеливания нагрузки с использованием кластерных и GRID-технологий. Также интересным направлением исследований является совершенствование стратегий управления коммутацией запросов [2, 3].

В совокупности работы исследователей в описанных выше направлениях формируют достаточное представление о методиках построения распределенных систем и применении технологий параллельных вычислений, позволяющих осуществлять анализ потоковых данных в режиме реального времени [4, 5]. Однако моделирование и анализ систем обработки информации обычно проводятся с точки зрения оптимального управления и распределения запросов в вычислительном комплексе в целом. В исследованиях недостаточное внимание уделяется эффективному взаимодействию источников и потребителей потоковой информации: уменьшению потокового трафика для снижения нагрузки на систему, разделению функций обработки сигналов для увеличения скорости обработки.

Задачи подобного рода актуальны, например, при управлении удаленными распределенными мехатронными системами в случаях, когда используют не выделенные каналы связи, а сети общего доступа.

Постановка задачи

В соответствии с описанным выше актуальной является задача повышения эффективности функционирования преобразователей потоковой информации в оперативных автоматизированных информационно-измерительных системах реального и приближенного к реальному времени.

Одно из направлений по совершенствованию технической базы средств вычислительной техники и элементов систем управления заключается в повышении их производительности и эффективном использовании пропускной способности каналов передачи. Этого можно добиться применением компрессии. Для сжатия разработано множество общих и специализированных алгоритмов для определенных типов данных, которые содержат в основе различные математические аппараты: преобразование Фурье, разложение в ряд Тейлора, метод Лагранжа, фрак-

тальный алгоритм, алгоритмы Лемпела—Зива—Велча (LZW) и Хаффмана, арифметическое кодирование и т. д. Наиболее перспективным направлением, по мнению авторов, является теория вейвлетов. Этот математический аппарат предлагается использовать для обработки потоковой информации в адаптивных системах преобразования данных.

Алгоритмы компрессии потоковой информации могут быть реализованы тремя способами: аппаратно, программно, программно-аппаратно. При разработке системы, предназначенной для решения рассматриваемой в настоящей работе задачи, использован программно-аппаратный подход. В настоящей статье рассматриваются только технические аспекты построения программной части комплекса. Аппаратная часть предлагаемой прикладной системы описана в работе [6] и базируется на использовании программируемых логических интегральных схем (ПЛИС) с возможностью динамического реконфигурирования (тип FPGA). Архитектурные особенности ПЛИС приспособлены для реализации таких операций, как умножение, свертка и т. п.

При этом в качестве критерия оценки эффективности функционирования используется общее время полного цикла обработки информации, которое включает следующие составляющие: 1) обработка на стороне преобразователя; 2) компрессия на стороне преобразователя; 3) транслирование средствами телекоммуникаций; 4) декомпрессия на стороне блока управления; 5) обработка на стороне блока управления. Сокращение времени передачи данных между блоками вторичного преобразователя и управления влечет увеличение длительности стадий компрессии и декомпрессии информации. Это влияние можно компенсировать, передав часть из набора функций обработки информации на сторону вторичного преобразователя. Последнее также полезно при увеличении числа клиентов (вторичных преобразователей), обрабатываемых одним блоком управления.

Предложенный программный комплекс информационно-измерительной системы основывается на клиент-серверной архитектуре. Отличительной особенностью системы является возможность перераспределения части из набора функций между клиентской и серверной сторонами обмена информацией.

Для создания программного комплекса необходимо решить перечисленные далее задачи.

1. Синтезировать модульную структуру программного комплекса.
2. Разработать схему иерархии и взаимодействия классов и объектов, составляющих систему.
3. Описать механизмы ввода/вывода информации, а также средства управления и настройки системы.
4. Разработать ядро системы, обеспечивающее ее функциональность и легкость создания и добавления новых компонентов.
5. Описать базовые функции для работы с потоковыми данными.

6. Разработать специализированные функции анализа информации, применяемые для решения задач определенного характера.

7. Сформировать схему логики изменения внутренних состояний программного комплекса.

Далее рассмотрим общую схему применения адаптивной системы обработки потоковой информации.

Схема адаптивной системы управления, сбора и анализа данных

На рис. 1 представлена структура комплекса, в основе которого содержится работа с потоковой сигнальной информацией.

Комплекс состоит из распределенной системы конечного числа элементов измерительных узлов (ИЗУ1...ИЗУ n), собирающих сигнальную информацию измерительного, регулирующего или управляющего характера средствами чувствительных элементов (датчики, сенсоры) с конечного числа объектов наблюдения (О1...О n). Измерительные узлы осуществляют преобразование сигналов датчиков и формируют потоковый контент (например, видеопоток) в реальном времени, передаваемый по информационным каналам. Такие узлы могут быть географически разнесены на значительные расстояния. Однако с использованием коммуникационных средств связи, например, локальной сети или сети Интернет, по которым транслируются информационные каналы, данные передаются в адаптивную систему управления потоковой информацией (АдСУ ПИ). Эта система осуществляет окончательное преобразование массивов данных, их анализ и, при необходимости, может содержать процедуры и функции экспертной системы для поддержки принятия решений.

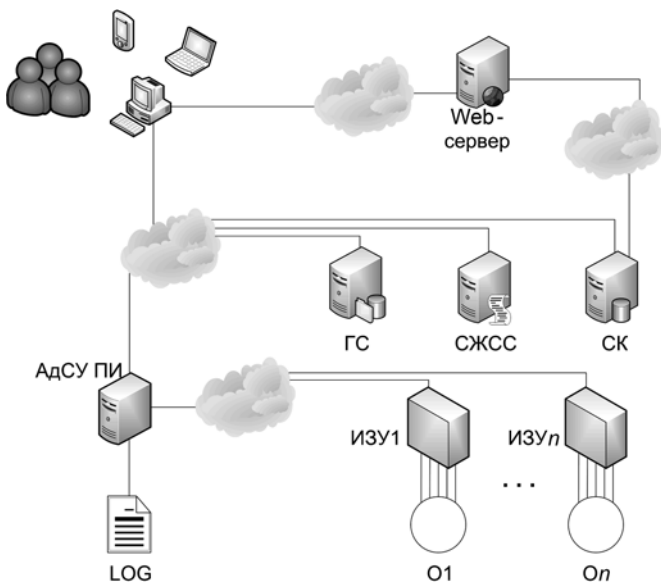


Рис. 1. Схема адаптивной системы управления, сбора и анализа потоковой информации

Управляющая система сохраняет преобразованные по заданным правилам данные и выявленные из них локальные характеристики исследуемых сигналов в базе данных (БД), которая содержится на сервере контента (СК). Кроме этого, при необходимости, осуществляется резервное копирование данных через заданный интервал времени на "горячий" сервер (ГС).

В свою очередь, пользователи, операторы, менеджеры и другой персонал могут получить доступ к информации, хранящейся в БД, через Интернет-сервис, реализованный в виде web-интерфейса (web-сервер). Результаты обработки представляются пользователям в виде графической информации, таблиц статистики. В этом аспекте разрабатываемая информационная система обладает такими важными свойствами, как кросс-платформенность, масштабируемость, стандартность механизмов ввода/вывода. В частном случае такие элементы, как web-сервер, СК, ГС, могут располагаться на одном компьютере.

Настройка управляющей системы может осуществляться как на локальном компьютере в терминале командной строки (CLI), так и удаленно через Telnet- или SSH-соединения. Для повышения информативности анализа работы система обеспечивает сохранение файла с записями о состоявшихся событиях в хронологическом порядке (log file, LOG), регистрирующих как внешние события, так и протоколирование работы программы. Кроме этого, система управления потоковым контентом опционально может отправлять сообщения о происходящих в системе событиях в удаленный системный журнал (сервер журналирования системных событий, СЖСС).

Функциональная часть АдСУ ПИ базируется на клиент-серверной парадигме, при этом часть операций по анализу данных может быть передана на клиентскую сторону, реализуемую в измерительных блоках ИЗУ. Рассмотрим более детально механизм передачи потоковой информации между АдСУ ПИ и ИЗУ.

Схема потокового обмена данными

На рис. 2 показан механизм взаимодействия клиент-серверной архитектуры. Данные, которые сняты с датчиков преобразователя ИЗУ, подвергаются вейвлет-преобразованию (ВП), которое реализовано аппаратно использованием функциональных возможностей ПЛИС. В качестве ядра свертки используется определенная материнская вейвлет-функция, которая хранится в базе данных. Применение ВП позволяет, во-первых, сконцентрировать энергию сигнала в меньшем объеме информации, во-вторых, выделить множество локальных особенностей сигналов. Кроме этого, этап ВП сигналов может быть опущен, например, в случае, когда обработка данных сконцентрирована в АдСУ ПИ.

Далее, уменьшение потокового трафика для снижения нагрузки на систему достигается сжатием пре-

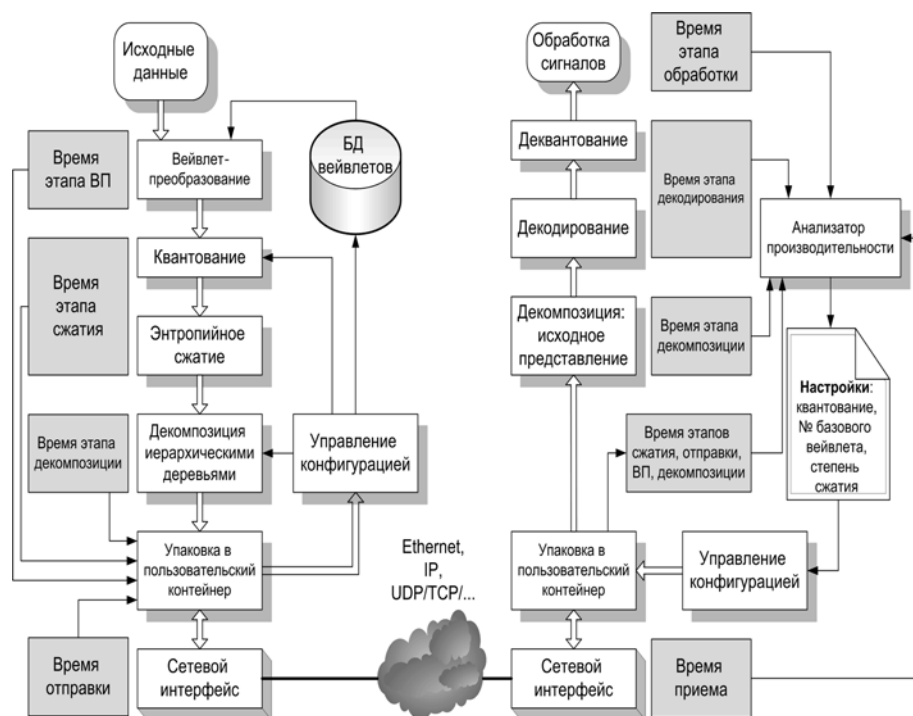


Рис. 2. Схема коммутации потоковой информации

образованной информации. Для компрессии применяется наиболее эффективный метод энтропийного сжатия без потерь — арифметическое кодирование. Эффект сжатия достигается применением квантования для свертки. Метод квантования также может быть изменен в процессе работы системы. Этапы компрессии также осуществляются средствами ПЛИС, однако могут отсутствовать в конфигурации.

Сжатая таким образом сигнальная информация подвергается декомпозиции и представлению в виде связанных иерархических деревьев. При этом данные аппроксимируются минимальным числом байт, которые группируются в начале последовательности, а затем эта аппроксимация иерархически детализируется. Для такого типа представления восстановленная информация всегда будет максимально хорошего качества для заданного размера байт памяти. Это применимо для передачи по каналам связи с различной пропускной способностью и степенью потерь. Кроме этого, такой поток байт может быть прерван в заданный момент, что регулирует произвольную степень компрессии, прерывание также может быть задано из конфигурации. Этот этап преобразования также может отсутствовать.

Далее сигнальная потоковая информация упаковывается в пользовательский протокол. Вместе с ней упаковывается и статистическая информация: показатели скорости обработки данных на каждом из этапов.

Пользовательский контейнер передается по коммуникационной сети кадрами (фрейм за фреймом) с использованием стандартных транспортных протоколов. При использовании прогрессирующего мето-

да передачи на приемной стороне уровень детализации информации будет увеличиваться последовательно (итерационно) с каждым следующим принятым блоком данных (фреймом).

На стороне АдСУ ПИ переданные данные, в том числе статистика скорости обработок на различных этапах в ИЗУ, распаковываются из пользовательского контейнера. После этого восстанавливается исходная последовательность байт — оригинальное представление информации. Затем данные декодируются и отправляются на дальнейшую обработку.

В параллельном потоке исполнения собственные операции реализует анализатор производительности. Для принятия решения компонент использует набор показателей о скорости выполнения этапов обработки информации в измерительном блоке (время этапов ВП, энтропийного сжатия, декомпозиции иерархическими

деревьями), в системе управления потоковой информации (время этапов высокоуровневой обработки сигналов, декодирования сжатий информации, восстановления исходной последовательности данных). Кроме этого, собирается информация о скорости передачи по каналам связи с использованием текущего транспортного протокола как разница между показателями времени отправки пакета и времени приема пакета. После анализа описанных показателей формируется файл с новой конфигурацией, который включает выбор метода квантования, объема данных для хранения свертки после изменения вида представления данных в памяти, номера материнского вейвлета, который будет использоваться для ВП сигнальной информации.

Входным воздействием анализатора производительности, кроме показателей времени выполнения этапов обработки очередного пакета, являются и собственными данными. Компонент анализатора производительности представляет гибридный нечеткий регулятор (ГНР), который обладает определенной степенью латентности по отношению ко входному воздействию. Для этого используется ряд буферов, применяемых для сохранения определенного количества последних пришедших пакетов, в том числе тех, время обработки которых превышает установленный порог. Кроме этого, используется медианный фильтр для регулирования чувствительности срабатывания ГНР. Глубина буферов и чувствительность срабатывания задаются в настройках системы. Компонент ГНР состоит из четырех элементов нечеткой логики и модуля, реализующего генетический

алгоритм поиска оптимальных параметров сжатия потоковой информации. В качестве данных, используемых для поиска характеристик компрессии, применяются сохраненные в буферах пакеты, время обработки которых превышает пороговое значение при текущих значениях параметров компрессии. Активацию этапа генетического поиска определяет значение выхода одного из блоков нечеткой логики. Другие блоки нечеткой логики применяются для определения граничных условий генетического поиска, выбора нового типа вейвлета из доступных в системе, определения необходимости смены транспортного протокола. Наконец, ГНР принимает одно из двух решений: оставить текущую конфигурацию без изменений или принять новую конфигурацию в качестве рабочей. Более детальное описание работы системы управления выходит за рамки настоящей статьи и ей будет посвящена отдельная работа.

Компонент управления конфигурацией формирует управляющий пакет из новой конфигурации и пересылает по каналу связи на измерительный узел. В узле новая конфигурация применяется, последующая сигнальная информация обрабатывается в соответствии с ней.

Теперь рассмотрим более детально предлагаемую функциональную схему адаптивной системы управления потоковой сигнальной информацией.

Функциональная структура адаптивной системы управления

Проектирование масштабируемого, надежного адаптивного программного комплекса для коммуникационных устройств вычислительной техники является сложной задачей. Такой программный продукт должен удовлетворять строгим требованиям по качеству выполнения функциональных операций, критически важных сервисов коммуникации между узлами. Теоретическая и методологическая базы проектирования схемы предлагаемого программного комплекса описаны в работах [7, 8].

Программная реализация построена с использованием преимуществ объектно-ориентированного подхода по модульному принципу. Каждый модуль (компонент) реализует строго определенные функции, при необходимости обращаясь через интерфейс к другим компонентам для получения или преобразования данных. Отметим, что система может быть собрана, при необходимости, без некоторых модулей, при этом упрощаются разработка и интеграция новых функций в виде новых модулей. Для каждого класса в системе создается абстрактный класс, единственное назначение которого — определить общий интерфейс для обращения к объекту. Это позволяет компонентам не заботиться о конкретных типах используемых данных или логике преобразования, при этом работа с данными осуществляется "более абстрактно" — программирование выполняется в соответствии с интерфейсом, а не с реализацией. Таким образом, реализация компонента может быть

полностью изменена с сохранением при этом интерфейса.

Адаптивная система управления потоковой информацией состоит из четырех блоков функций и блока контейнеров (рис. 3). Каждый функциональный блок обеспечивает функционирование комплекса на собственном уровне. На структурной схеме системы, которая иллюстрирует взаимодействие абстрактных компонентов, указана литера I (интерфейс класса), таким образом формируется спецификация связи между компонентами. Рассмотрим кратко назначение и состав каждого блока.

Ключевым, обеспечивающим функционирование системы, является блок *системных функций*. Программный комплекс системы управления является многопоточным приложением. Каждый компонент имеет собственный поток исполнения, который может быть запущен один раз. Далее осуществляется обращение к существующему объекту, что обеспечивается применением паттерна проектирования Singleton. Первым запускается поток Init, из него осуществляется создание и запуск на исполнение остальных модулей.



Рис. 3. Функциональная структура системы управления

Каждый модуль состоит из трех частей: спецификатор (Specifier), который описывает отображение конфигурации, позволяет регистрировать команды для настройки компонента и создает процесс обработчика; конфигурационные команды (ConfigurationCommands); основной обработчик (Handler). При этом конфигурационные команды необязательны, так как некоторые компоненты могут не иметь возможности изменения конфигурации. Запуск компонентов осуществляется в две стадии:

1) выполнение спецификатора, в результате создается новый поток, если он не был запущен, регистрируются в системе новые команды, выполняющие конфигурацию;

2) запуск функции обработчика в потоке выполнения.

Таким образом, первоначально Init осуществляет создание и инициализацию потоков выполнения для каждого компонента (стадия 1). После успешной инициализации всех компонентов в системе управления существует конечное число потоков, приостановленных и ожидающих начала выполнения. Далее Init переходит к стадии 2 инициализации, каждому потоку посылается сигнал запуска обработчика. После этого Init ожидает окончания всех дочерних потоков или сигнала завершения программы от системы (рис. 4).

Конфигурация представляет текстовый файл и применяется компонентом Configurator построчно (каждая строка представляет одну команду), используя компонент CИ для обработки команд.

Еще одним функциональным блоком являются *утилиты*. Каждый модуль этого блока позволяет расширить функциональные возможности программного комплекса. Например, добавляет возможность сохранения системного журнала событий (Log) или позволяет периодически сохранять резервную копию данных на "горячем" сервере (HotServer).

Контейнеры позволяют обмениваться блоками данных строго определенного формата между компонентами. Например, контейнер для хранения UDP-пакета, с одной стороны, позволяет хранить модифицируемые данные перед отправкой пакета и формировать фрейм из данных по правилам спецификации протокола, с другой стороны, позволяет выполнить декомпозицию принятого из сети фрейма и получить доступ к полезной нагрузке.

Описанное выше иллюстрирует работу ядра системы. Далее более детально рассмотрим функции программного комплекса для работы с потоковыми данными.

Характеристика базовых функций ядра комплекса

Общие методы работы с потоковой сигнальной информацией имплементированы в модулях, формирующих блок *базовых функций*.

Вне зависимости от специфики области применения программного продукта, массивы потоковых данных проходят одинаковые этапы обработки. Эти этапы базовой обработки и управления потоковой информацией приведены на рис. 2.

Как описано выше, функциональные возможности, осуществляющие ВП, квантование, энтропийное сжатие и декомпозицию свертки связными иерархическими деревьями на измерительном узле ИЗУ, реализуются аппаратными средствами ПЛИС типа FPGA и в программной реализации. Причем конфигурация, которой динамически программируется плата в процессе работы, учитывает характерные особенности работы с устройствами класса встраиваемых систем. В рамках настоящей статьи не рассматривается собственно программная реализация блока ИЗУ. Подробно будут рассмотрены функции этапов обработки потоковых данных, реализованные в АдСУ ПИ.

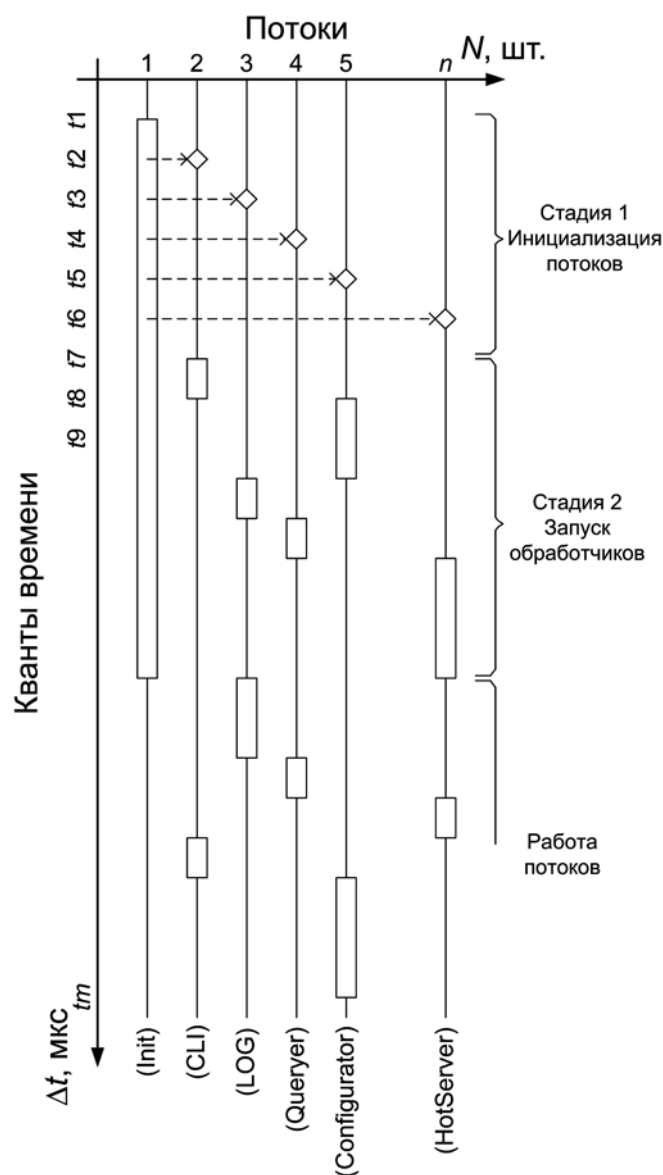


Рис. 4. Временная диаграмма потоков системы управления

После извлечения полезной нагрузки из протокола транспортного уровня и распаковки потоковой информации из контейнера в работу вступает компонент, изменяющий порядок байт представления в памяти (IDataSequenceViewer). Его основное предназначение — трансформация между прогрессирующим методом расположения байт. При этом исходный блок данных аппроксимируется минимальным числом байт, которые группируются в начале последовательности, а затем эта аппроксимация иерархически детализируется байтами "хвоста" последовательности и исходным методом представления и обратно. С изменением представления материала компонент также возвращает характеристику, описывающую длительность выполнения преобразования. Эта и другие характеристики учитываются далее компонентом анализа производительности (IPerformance).

Следующий функциональный элемент — декодер (IDecompressor), который восстанавливает исходный сигнал. Методом компрессии/декомпрессии выбрано арифметическое кодирование. Однако если сигнальная информация закодирована на ИЗУ с потерями, восстановленный образ не будет идентично соответствовать исходному образу сигналов. При этом восстановленный вариант имеет искажения, для уменьшения которых применяется процедура деквантования (IQuantizer).

Восстановленный, приближенный образ свертки может быть использован непосредственно специализированными функциями анализа потоковой информации либо подвержен обработке вейвлет-преобразованием образа средствами компонента IWaveletTool.

Кроме описанных процедур управления потоковыми данными, к блоку базовых функций относится также механизм сбора фреймов потокового вещания (ICollector) из списка настроенных адресов ИЗУ. Управлением изменения конфигурации на удаленном устройстве руководит модуль IQueuer, который упаковывает новую конфигурацию в специальный управляющий пакет и осуществляет запрос к конкретному ИЗУ для применения указанной конфигурации.

Далее в качестве примера рассмотрим реализацию специализированных функций анализа потоковой информации для решения конкретных задач из различных предметных областей.

Специализированные функции анализа потоковой информации

К *специализированным функциям* относится набор классов, реализующих обработку сигнальной информации, характерную для конкретной задачи. Такая обработка осуществляется на более высоком уровне. Функции, позволяющие осуществлять обработку и анализ строго определенных потоковых данных, обычно организуются в виде отдельной библиотеки. Например, процедуры, позволяющие обрабатывать видеопотоковые данные от стереоскопической системы технического зрения (ССТЗ) с лазерным ска-

нированием (выделение габаритных точек объектов, триангуляция, объединение полигонов, декомпозицию сцены на отдельные объекты и т. д.), составляют одну библиотеку; процедуры для анализа графиков динамограмм (расчет напряжения на плунжере, вычисление напряжения на штоке, определение точек разрывов и т. д.) — другую библиотеку; процедуры, осуществляющие обработку потокового контента (определение контрольных точек и положения глаза, выделение границы зрачка, определение площади зрачка и т. д.), — третью библиотеку.

Таким образом, собрав основную часть системы с использованием различных библиотек, получаем программный комплекс обработки, хранения и анализа потоковой информации в различных областях применения (научных, технических, медицинских).

Рис. 5 иллюстрирует результат обработки потокового контента в каждой из описанных выше областей применения с использованием одной из трех предложенных библиотек.

На рис. 5, *а* представлен результат сканирования поверхности стереоскопической системой с лазерной подсветкой (рельеф поверхности), для отображения которого проведен расчет трехмерных координат каждой точки и их триангуляция с выделением полигонов. На рис. 5, *б* представлен результат обработки потокового видео с устройства детектирования реакции зрачка человеческого глаза на внешние раздражители в виде вспышек света, на изображении определено положение зрачка глаза, выделен контур окружности. После этого возможно определение площади зрачка и скорости ее изменения. На рис. 5, *в* представлен результат сбора и анализа множества показателей датчиков, установленных на оборудовании в сочетании со штанговым глубинным насосом, применяемым в нефтедобыче из малобетонных месторождений. Анализ динамограмм в сочетании с дополнительными измеряемыми характеристиками используется для диагностики состояния геодезического и добывающего оборудования.

Из характера областей применения описанных задач следует, что выделить общие процедуры высокоуровневой обработки потоковой информации, которые применяются в определенной прикладной области, весьма сложно. Поэтому заранее характеризовать каждую возможную функцию не имеет смысла. Набор функций библиотеки должен разрабатываться с учетом собственной специфики.

Таким образом, описаны функции ядра системы, базовых процедур работы с потоковой информацией и специализированными методами обработки полученных массивов данных на последующем уровне в цикле анализа информации. В завершение рассмотрим диаграмму машины состояний алгоритма, обеспечивающего обработку потока данных.

Машина состояний алгоритма работы программного комплекса

Адаптивная система управления сбором, сохранением и анализом потоковой сигнальной информации представляет программный комплекс со сложной внутренней взаимосвязью компонентов. В настоящем подразделе рассмотрим диаграмму только центральной части системы управления, отвечающей

за изменение внутренних состояний по этапам обработки потоковой сигнальной информации (рис. 6).

Из диаграммы следует несколько ключевых пунктов:

- анализ производительности обработки потоковой информации осуществляется параллельно, в отдельном потоке исполнения;
- для анализа используется комплексный критерий, включающий показатели скорости обработки потоковых данных на этапах в ИЗУ (ВП, сжатие,

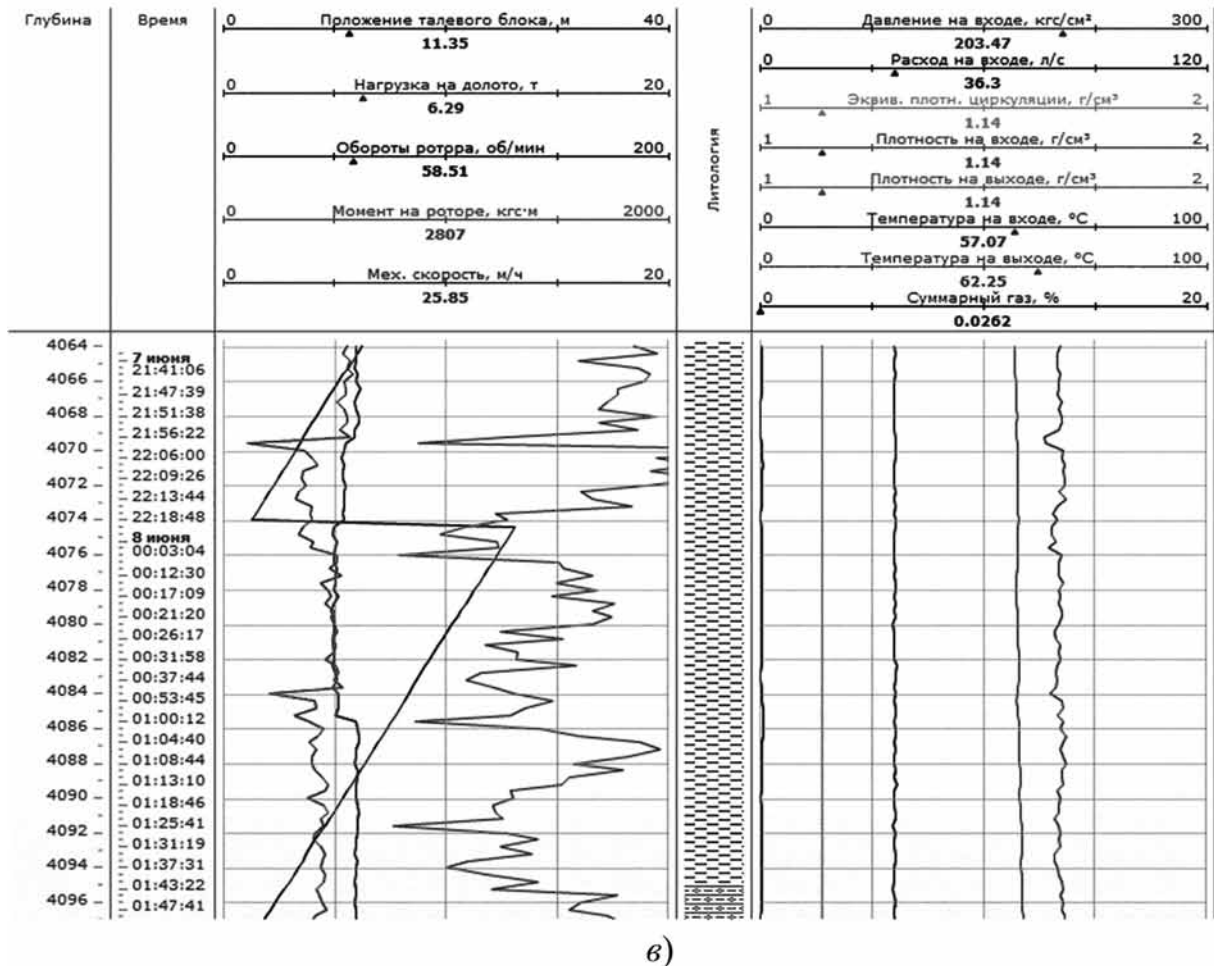
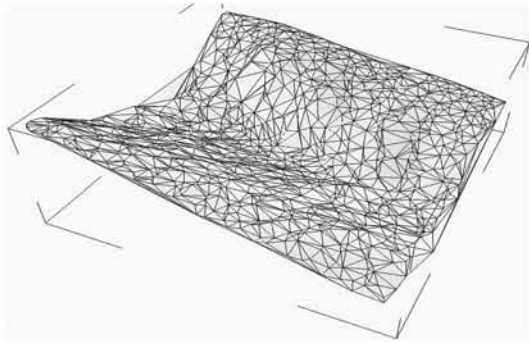


Рис. 5. Примеры, иллюстрирующие результат использования системы в различных областях применения:

а — рельеф поверхности, построенный по пространственным точкам в результате сканирования ССТЗ с лазерной подсветкой; б — выделение контура зрачка; в — сигнальная информация от датчиков на штанговом глубинном насосе

декомпозиция и представление в виде связанных иерархических деревьев), в АдСУ ПИ (обратная декомпозиция, восстановление приближенного образа свертки, высокоуровневая обработка сигнальной информации) и времени передачи по коммуникационной сети;

- существует несколько точек входа в операцию изменения конфигурации.

Типовой маршрут характеризует остановку текущей передачи данных по информационному каналу и открытие нового сеанса с измененными значениями конфигурационных характеристик. Адаптивный



Рис. 6. Маршрутная схема процесса управления потоковой информацией

маршрут — закрытие сеанса не осуществляется, конфигурация изменяется в принудительном режиме.

Заключение

Предложенная адаптивная система преобразования потоковой информации обеспечивает возможность подстраиваться под достаточно широкий набор данных с различными статистическими характеристиками, а не специализироваться на определенном виде.

Это требование осуществимо применением конечного набора вейвлет-функций, каждая из которых позволяет более эффективно обрабатывать поток информации определенного типа. Также при изменении характеристик передающей среды система обеспечивает возможность динамически изменять степень сжатия потоковой информации, осуществляя реконфигурацию блоков компрессии и транслирования.

Таким образом, адаптивную информационно-измерительную систему можно применять в различных областях, например, в задачах распознавания объектов на основе потоковой информации, поступающей от стереоскопической системы технического зрения, от комплексов мониторинга и диагностики состояния штангового глубинного насоса, а также от устройств детектирования реакции зрачка человеческого глаза на внешние раздражители в виде вспышек света.

Список литературы

1. Красько Н., Мильцина А. Адаптивное потоковое вещание — в поисках единого решения // Теле-Спутник. 2012. № 7. С. 70—73.
2. Заяц О. И., Заборовский В. С., Мулюха В. А., Вербенко А. С. Управление пакетными коммутациями в телематических устройствах с ограниченным буфером при использовании абсолютного приоритета и вероятностного выталкивающего механизма. Часть 1 // Программная инженерия. 2012. № 2. С. 22—28.
3. Заяц О. И., Заборовский В. С., Мулюха В. А., Вербенко А. С. Управление пакетными коммутациями в телематических устройствах с ограниченным буфером при использовании абсолютного приоритета и вероятностного выталкивающего механизма. Часть 2 // Программная инженерия. 2012. № 3. С. 21—29.
4. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
5. Топорков В. В. Модели распределенных вычислений. М.: ФИЗМАТЛИТ, 2004. 320 с.
6. Большаков А. А., Лачугин Д. В., Лобанов В. В. Создание архитектуры программно-аппаратного модуля вторичного преобразователя информации для адаптивной системы обработки информации // Научные технологии и инновации: сб. докл. Междунар. науч.-практ. конф. Часть 6. Белгород: Изд-во БГТУ, 2014. С. 122—126.
7. Goma H. Software Modeling and Design: UML, Use Cases, Patterns & Software Architectures. Cambridge University Press, 2011. 592 p.
8. Utas G. Robust Communications Software: Extreme Availability, Reliability and Scalability for Carrier-Grade Systems. John Wiley & Sons Ltd, 2005. 352 p.

A. A. Bolshakov, Professor, e-mail: aabolshakov57@gmail.com,
St. Petersburg Polytechnic University,

D. V. Lachugin, Postgraduate Student, Assistant, e-mail: lachuginm@mail.ru,

V. V. Lobanov, Associate Professor, e-mail: lobanovv@gmail.com,
Yuri Gagarin State Technical University of Saratov

Design of Software of an Adaptive System of Control, Collection and Analysis of Streaming Data

A scheme of adaptive information-measuring system of management, collection and analysis of signal data flow is proposed. A major feature of adaptive system is the ability to dynamically change the functional configuration to reduce the total time of processing of streaming information. Also some technical problems of synthesis of software server and client sides based on the proposed reconfigurable computing system structure are described in this paper.

Keywords: adaptive control system, streaming data, reconfigurable computing system, software system

References

1. Kras'ko N., Mil'cina A. Adaptivnoe potokovoe veshhanie — v poiskah edinogo reshenija. *Tele-Sputnik*. 2012. N. 7. P. 70—73 (in Russian).
2. Zajac O.I., Zaborovskij V.S., Muljuha V.A., Verbenko A.S. Upravlenie paketnymi kommutacijami v telematicheskikh ustrojstvah s ogranichenym buferom pri ispol'zovanii absoljutnogo prioriteta i verojatnostnogo vytalkivajushhego mehanizma. Part 1. *Programmna-ya ingeneria*. 2012. N. 2. P. 22—28 (in Russian).
3. Zajac O.I., Zaborovskij V.S., Muljuha V.A., Verbenko A.S. Upravlenie paketnymi kommutacijami v telematicheskikh ustrojstvah s ogranichenym buferom pri ispol'zovanii absoljutnogo prioriteta i verojatnostnogo vytalkivajushhego mehanizma. Part 2. *Programmna-ya ingeneria*. 2012. N. 3. P. 21—29 (in Russian).
4. Voevodin V.V., Voevodin V.I.V. *Parallel'nye vychislenija*. SPb.: BHV-Peterburg, 2002. 608 p. (in Russian).
5. Toporkov V.V. *Modeli raspredelennyh vychislenij*. M.: FIZMATLIT, 2004. 320 p. (in Russian).
6. Bol'shakov A.A., Lachugin D.V., Lobanov V.V. Sozdanie arhitektury programmno-apparatnogo modulja vtorichnogo preobrazovatelya informacii dlja adaptivnoj sistemy obrabotki informacii. *Nauchemkie tehnologii i innovacii*: sb. dokl. Mezhdunar. nauch.-prakt. konf. Part 6. Belgorod: Izd-vo BGTU, 2014. P. 122—126 (in Russian).
7. Goma H. *Software Modeling and Design: UML, Use Cases, Patterns & Software Architectures*. Cambridge University Press, 2011. 592 p.
8. Utas G. *Robust Communications Software: Extreme Availability, Reliability and Scalability for Carrier-Grade Systems*. John Wiley & Sons Ltd, 2005. 352 p.

И. О. Жаринов, д-р техн. наук, доц., зав. каф., e-mail: igor_rabota@pisem.net, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (НИУ ИТМО), руководитель учебно-научного центра, Санкт-Петербургское ОКБ "Электроавтоматика" им. П. А. Ефимова,
О. О. Жаринов, канд. техн. наук, доц., Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП)

Способ программного управления яркостью бортового средства индикации в режиме ночного полета летательного аппарата

Рассмотрена задача программного управления яркостью бортового средства индикации, выполненного на базе жидкокристаллической панели, в режиме ночного полета летательного аппарата. Предложен новый способ управления яркостью изображения. Данный способ основан на введении масштабного коэффициента и пропорциональном программном изменении цифровых кодов компонентов основных цветов. Способ позволяет регулировать яркость изображения в диапазоне от 0,1 до 100 % уровня физической яркости, установленного средствами ручной регулировки. Основное достоинство предлагаемого способа заключается в сохранении координат цветности индицируемых элементов изображения и сохранении неравномерности яркости экрана во всем диапазоне регулировки. Приведены графики зависимости коэффициента пропуска жидкокристаллической панели от масштабного коэффициента изменения кодов основных цветов. Даны результаты измерений яркости в восьми цветах, применяемых в авионике, при различных значениях масштабного коэффициента.

Ключевые слова: средства индикации, авионика, управление яркостью, масштабирование, координаты цветности

Введение

В состав бортовой системы индикации современных пилотируемых летательных аппаратов входят многофункциональные цветные индикаторы (МФЦИ) [1], экран которых выполнен на базе плоской жидкокристаллической (ЖК) панели. МФЦИ предназначены для индикации пилоту пилотажно-навигационных параметров, геоинформационных данных (цифровой карты местности), состояния общесамолетного оборудования и др.

Информация, индицируемая на экране МФЦИ, используется пилотом во всех фазах выполнения полета (от взлета до посадки) для управления летательным аппаратом (ЛА). Одним из режимов полета ЛА является режим ночного полета.

Опыт разработки и анализ результатов эксплуатации МФЦИ показывают, что работа МФЦИ в режиме ночного полета ЛА имеет ряд особенностей [2–4], обусловленных необходимостью обеспечения повышенных характеристик восприятия пилотом информации при малом уровне яркости ЖК экранов. По сложившейся в авиационной промышленности

практике разработчики МФЦИ стремятся обеспечить высокий уровень яркости свечения ЖК экрана, ориентируясь на основной — дневной — режим эксплуатации ЛА в условиях воздействия на экран прямой солнечной засветки, и не уделяют при этом должного внимания алгоритмам управления яркостью ЖК экрана в режиме ночного полета ЛА.

Между тем замечено, что индикаторам МФЦИ с ЖК экранами, выполненными как на базе газоразрядного подсвета, так и на базе светодиодного подсвета, при малой яркости изображения (до 20 кд/м²) присущи повышенные уровни неравномерности яркости свечения экрана (до 50 %), что вызывает затруднение в восприятии пилотом индицируемой информации, а для газоразрядных ЖК экранов наблюдается также мерцание изображения на экране. Таким образом, летный состав вынужден принудительно увеличивать ручными органами управления яркость свечения экрана для обеспечения стабильной и приемлемой пилоту работы МФЦИ в условиях ночного полета [5–8].

Повышенная яркость свечения ЖК экрана в режиме ночного полета ЛА (в темноте) также повышает

нагрузку на зрительный аппарат пилота и в ряде случаев (для "прозрачной" кабины ЛА класса МиГ, Су и пр.) является недопустимой вследствие демаскирования положения ЛА в воздушном пространстве.

В связи с этими обстоятельствами актуальной является задача исследования и разработки новых принципов управления яркостью изображения на экране МФЦИ для обеспечения стабильной и качественной работы бортового средства индикации в режиме ночного полета ЛА без ухудшения эргономических показателей качества восприятия изображения.

Способ управления яркостью изображения на ЖК экране МФЦИ

Общепринятый способ управления яркостью ЖК экрана предполагает воздействие в ручном или автоматическом режиме на параметры (амплитудное значение, скважность широтно-импульсной модуляции и др.) сигнала управления яркостью. Изменение параметра сигнала управления яркостью приводит к изменению амплитуды и/или формы управляющего воздействия на источник подсвета ЖК экрана (газо-разрядная лампа) и к изменению яркости свечения экрана. Изменение яркости свечения экрана, в свою очередь, приводит к изменению контрастных характеристик восприятия изображения.

Для такого аппаратного способа управления яркостью характерной особенностью является наличие в графическом контроллере МФЦИ цифро-аналоговых преобразователей, формирующих заданную циклограмму управления подсветом ЖК экрана. Цифровой код управления формируется микроконтроллером или элементом программируемой логической интегральной схемы на основе результатов обработки сигналов от кнопочных органов управления МФЦИ. При этом дискретность изменения уровней яркости определяется разрядностью цифро-аналогового преобразователя. В условиях реального полета предпочтительно использовать не более 8–10 фиксированных уровней яркости.

Новый принцип управления яркостью изображения на ЖК экране МФЦИ предполагает не физическое воздействие на источник заднего или бокового подсвета ЖК экрана, а программное воздействие на строчные и столбцовые драйверы, отвечающие за управление жидкими кристаллами. Важно заметить, что предлагаемый программный способ управления яркостью применим для ЖК матриц, так как изменение яркости свечения экрана осуществляется за счет изменения коэффициента пропускания матрицы в заданном цвете, достигаемого путем поворота жидких кристаллов.

Опытным путем замечено, что при фиксированном уровне яркости ЖК экрана, например, 600 кд/м^2 , установленном органами управления яркости в ручном режиме, программным способом может быть изменена яркость изображения, воспринимаемая пилотом, вплоть до визуального ощущения полного погасания экрана без изменения амплитудно-временных па-

раметров управляющего воздействия на источник подсвета. Основной положительный эффект от такого программного способа управления яркостью заключается в инвариантности координат цветности элементов изображения, индицируемых на экране МФЦИ, по отношению к параметрам сигнала управления кристаллами ЖК экрана.

Программный способ управления яркостью заключается в следующем. Как известно [1], пиксель ЖК экрана состоит из трех кристаллов, каждый из которых имеет возможность вращения вдоль вертикальной оси и отвечает за коэффициент пропускания света в основных цветах (*Red* (красный) — *R*, *Green* (зеленый) — *G*, *Blue* (синий) — *B*). Таким образом, при наличии за ЖК экраном источника света белого спектра свечения каждый жидкий кристалл выделяет из спектра белого цвета свою часть спектра цвета (красную, зеленую или синюю) и управляет коэффициентом пропускания η света в этом цвете: $\eta = L_{\text{ц}}/L_{\text{п}} \cdot 100 \%$, где $L_{\text{ц}}$ — яркость изображения в цвете, наблюдаемая пилотом; $L_{\text{п}}$ — яркость заднего или бокового подсвета ЖК экрана в белом цвете.

В совокупности компоненты красного, зеленого и синего цветов при малых линейных размерах пикселя ЖК экрана воспринимаются наблюдателем как элемент цветного изображения. Совокупность элементов изображения на экране МФЦИ — бортовой индикационный кадр (рис. 1, см. вторую сторону обложки).

Управляющие коды *RGB* определяют угол поворота жидких кристаллов в основных цветах и влияют на значение коэффициента η . Нулевые значения кода *RGB* = {0, 0, 0} соответствуют индикации на ЖК экране пикселя черного цвета и ориентируют жидкие кристаллы таким образом, что коэффициент пропускания стремится к минимуму: $\eta \rightarrow \min, \eta \ll 1 \%$. Максимальные значения кодов *RGB* = {255, 255, 255} соответствуют индикации на ЖК экране пикселя белого цвета и ориентируют жидкие кристаллы таким образом, что коэффициент пропускания экрана $\eta \approx 100 \%$.

Результирующие координаты цвета каждого пикселя, воспринимаемого наблюдателем, получаются из десятичных кодов *RGB* в соответствии с законом смешения цветов Грассмана [9, 10]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (1)$$

где *X*, *Y*, *Z* — компоненты цвета в системе *XYZ* цветового треугольника Максвелла; $X_r, X_g, X_b, Y_r, Y_g, Y_b, Z_r, Z_g, Z_b$ — компоненты цвета, определенные Международной комиссией по освещению и используемые в качестве эталона для точного стандарта определения цвета; *R*, *G*, *B* — десятичный код цвета компонентов основных цветов в системе *RGB*. Компоненты X_r, Y_r, Z_r определяют правило преобразования кода *RGB* для эталонного значения красного цвета, компоненты X_g, Y_g, Z_g и X_b, Y_b, Z_b — для зеленого и синего цветов соответственно.

Переход от значений координат цвета XYZ треугольника Максвелла к (x, y) -координатам цветности на XY -плоскости и обратно осуществляется по следующим [9] формулам:

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad (2)$$

$$X = \frac{xY_{xy}}{y}, \quad Y = Y_{xy}, \quad Z = (1 - x - y) \frac{Y_{xy}}{y}, \quad (3)$$

где Y_{xy} — компонент относительной яркости, $Y_{xy} = Y$.

Анализ выражений (1)–(3) показывает, что линейное изменение значений десятичных кодов RGB , т. е. $\{kR, kG, kB\}$, где k — произвольное целое число, такое что $0 \leq kR \leq 255$, $0 \leq kG \leq 255$, $0 \leq kB \leq 255$, не приводит к изменениям значений (x, y) -координат цветности. Изменяется только значение компонента относительной яркости по следующему правилу: $Y_{xy} = kY$.

Таким образом, коды $RGB = \{1, 1, 1\}; \{25, 25, 25\}; \{100, 100, 100\}; \{255, 255, 255\}$, определяющие углы поворота жидких кристаллов в основных цветах и влияющие на значение коэффициента η , имеют одни и те же значения (x, y) -координат цветности, а значит, наименование цвета сохраняется при изменении k , и различаются только значениями компонента относительной яркости Y_{xy} .

По существу, введение масштабного коэффициента k позволяет регулировать насыщенность цветов одновременно всего индикационного кадра.

Следовательно, при наличии в составе МФЦИ фиксированной палитры цветов [11], заданной компонентами кодов RGB для каждого цвета и используемой для индикации значений пилотажно-навигационных параметров, способ управления яркостью ЖК экрана будет заключаться в изменении программными средствами масштабного коэффициента k , на который будет умножаться (делиться с округлением до ближайшего целого) значение кодов RGB .

В этом случае для режима ночного полета ЛА целесообразно при фиксированном значении яркости свечения ЖК экрана МФЦИ переключать органы управления ручной регулировки физической яркости на программное управление дискретным изменением значения масштабного коэффициента k .

Таким образом, алгоритм регулировки яркости следующий.

1. Формируется цветовая палитра МФЦИ, состоящая из наборов кодов RGB цветов, обладающих наилучшими визуальными характеристиками восприятия для человека. Методика формирования приведена в работе [2]. Формирование палитры осуществляется на этапе проектирования МФЦИ.

2. В процессе ночного полета ЛА осуществляется вывод индикационного кадра на ЖК экран МФЦИ с цветами, определенными в цветовой палитре.

3. При нажатии на органы управления ручной регулировки яркости выполняется пересчет с помощью формулы (1) кодов RGB в координаты цвета XYZ , пересчет с помощью формулы (2) координат цвета XYZ в (x, y) -координаты цветности, масшта-

бируется компонент относительной яркости Y_{xy} по правилу $Y_{xy} = kY$. Значение масштабного коэффициента k изменяется органами управления ручной регулировки яркости.

4. Выполняется обратное преобразование $XY_{xy}Z \rightarrow RGB$ по правилам (3), (1) для всей цветовой палитры, используемой в индикационном кадре.

5. Формируется индикационный кадр на ЖК экране МФЦИ с использованием цветовой палитры с масштабированными кодами RGB с измененной яркостью изображения. Осуществляется вывод изображения на ЖК экран.

Экспериментальные исследования программного способа управления яркостью

Для исследования программного способа управления яркостью ЖК экрана была проведена серия экспериментов. В эксперименте участвовали промышленные образцы МФЦИ, в которых установлены ЖК экраны четырех фирм-изготовителей с газоразрядным подсветом:

- SEXTANT, модель SMD-68;
- SHARP, модель LQ104V1DG11 с модулем подсвета 104UC1 WINSOR;
- NEC, модель NL6448AC33-29;
- SIEMENS, модель 75i.

Схема включения ЖК экранов реализована разработчиками МФЦИ.

Органами ручной регулировки яркости МФЦИ путем воздействия на управляющий сигнал подсвета устанавливалась определенная физическая яркость изображения ЖК экрана. Для МФЦИ с ЖК экраном SEXTANT в белом цвете установленная яркость составила 520 кд/м^2 ; для МФЦИ с ЖК экраном SHARP — 700 кд/м^2 ; для МФЦИ с ЖК экраном NEC — 335 кд/м^2 ; для МФЦИ с ЖК экраном SIEMENS — 635 кд/м^2 .

Далее органы управления регулировкой яркости программным способом переключались на управление изменениями масштабного коэффициента k . Значения десятичных кодов RGB , соответствующие дискретным значениям коэффициента k , используемым в эксперименте, приведены в табл. 1 для восьми цветов, входящих в цветовую палитру МФЦИ.

Для представленных в табл. 1 значений коэффициента k и соответствующих им кодов RGB проводились измерения яркости каждого цвета на МФЦИ в каждой градации с ЖК экранами различных изготовителей. Измерения яркости выполнены с использованием яркомера ЯРМ-3 в составе автоматизированного рабочего места [9]. Расчетным способом определялся коэффициент пропускания η . Результаты расчетов приведены на рис. 2 (см. вторую сторону обложки) для восьми отображаемых цветов (используются следующие цвета: белый (кривая черного цвета), красный (кривая красного цвета), зеленый (кривая зеленого цвета), синий (кривая синего цвета), пурпурный (кривая пурпурного цвета), голубой

Таблица 1

Значения десятичных кодов *RGB*, соответствующие дискретным значениям коэффициента *k* для восьми цветов

<i>k</i> , ед.	Цвет											
	Белый			Красный			Зеленый			Синий		
	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>
0	0	0	0	0	0	0	0	0	0	0	0	0
36	36	36	36	36	0	0	0	36	0	0	0	36
72	72	72	72	72	0	0	0	72	0	0	0	72
108	108	108	108	108	0	0	0	108	0	0	0	108
144	144	144	144	144	0	0	0	144	0	0	0	144
180	180	180	180	180	0	0	0	180	0	0	0	180
216	216	216	216	216	0	0	0	216	0	0	0	216
255	255	255	255	255	0	0	0	255	0	0	0	255

<i>k</i> , ед.	Цвет											
	Пурпурный			Голубой			Желтый			Коричневый		
	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>	<i>R</i>	<i>G</i>	<i>B</i>
0	0	0	0	0	0	0	0	0	0	0	0	0
36	36	0	36	0	36	36	36	36	0	36	12	0
72	72	0	72	0	72	72	72	72	0	72	24	0
108	108	0	108	0	108	108	108	108	0	108	36	0
144	144	0	144	0	144	144	144	144	0	144	48	0
180	180	0	180	0	180	180	180	180	0	180	60	0
216	216	0	216	0	216	216	216	216	0	216	72	0
255	255	0	255	0	255	255	255	255	0	255	85	0

Таблица 2

Измеренные значения яркости цветовых градаций

Цвет изображения	Изготовитель ЖК экрана в МФЦИ	Коэффициент <i>k</i> , ед.							
		0	36	72	108	144	180	216	255
		Яркость градаций, кд/м ²							
Белый	SEXTANT	4,3	30	52	81	120	165	260	520
	SHARP	8,6	19,3	64	174	351	472	610	700
	NEC	0,6	2,9	11,9	37,7	78	119	226	335
	SIEMENS	4,7	7,2	26,6	59,2	143,8	330	508	635
Красный	SEXTANT	3,7	6,9	8,6	12,4	17,6	25	36	61
	SHARP	6,6	8,9	22	47	107	137	168	183
	NEC	0,6	1	2,9	6,4	14,5	21,7	40,6	76
	SIEMENS	4,7	4,7	9,7	15,2	34,3	72	131	178
Зеленый	SEXTANT	4,3	26	44	70	99	130	194	278
	SHARP	7,8	14,5	46	125	244	335	396	427
	NEC	0,6	2,0	7,54	22,9	47,9	72,5	133	198
	SIEMENS	4,7	5,9	14	32,1	71,9	161	279	338
Синий	SEXTANT	4	13	22	32	46	58	75	93
	SHARP	6,6	9,2	22	46	107	122	153	153
	NEC	0,6	1,3	3,5	9	17,4	26	52	76
	SIEMENS	4,7	4,7	7,6	14	31,3	76,1	97	123

Цвет изображения	Изготовитель ЖК экрана в МФЦИ	Коэффициент k , ед.							
		0	36	72	108	144	180	216	255
		Яркость градаций, кд/м ²							
Пурпурный	SEXTANT	3,7	18	35	53	75	99	128	161
	SHARP	7,3	12	33	80	168	214	275	305
	NEC	0,6	1,6	5,2	14,5	29	46	93	137
	SIEMENS	4,7	5,5	13,5	27,1	63,5	152	228	296
Голубой	SEXTANT	4	22	51	77	110	157	186	220
	SHARP	7,5	12,7	33	75	168	260	305	366
	NEC	0,6	1,6	5,5	15	33,5	55	95	152
	SIEMENS	4,7	5,5	11,4	27,9	59,2	123	178	262
Желтый	SEXTANT	4	32	55	93	133	180	258	380
	SHARP	8,3	17,5	58	157	305	427	518	580
	NEC	0,6	2,3	9,3	29	61	93	174	260
	SIEMENS	4,7	6,8	22	46,5	106	233	415	508
Коричневый	SEXTANT	3,7	10	21	44	58	77	96	125
	SHARP	7	10	25	52	107	153	199	229
	NEC	0,6	1,2	3,2	7,8	16	26	46	91
	SIEMENS	4,7	5,1	9,7	18,2	38,9	85	152	195

(кривая голубого цвета), желтый (кривая желтого цвета), коричневый (кривая коричневого цвета)).

Анализ рис. 2 показывает, что программный способ управления позволяет регулировать коэффициент пропускания η ЖК экрана в пределах 0,1...100 % с сохранением значений (x, y) -координат цветности отображаемых элементов изображения. При этом все четыре промышленных образца МФЦИ с различными ЖК экранами обладают примерно одинаковыми характеристиками крутизны кривой для программного способа управления коэффициентом пропускания.

Таким образом, яркость изображения на экране МФЦИ регулируется программным способом в пределах 0,1...100 % от значения физической яркости, установленной органами управления яркости МФЦИ, без каких-либо воздействий на амплитудно-временные параметры сигнала управления подсветом.

Измеренные значения яркости для цветовых градаций, соответствующих выбранным значениям масштабного коэффициента k в восьми цветах, приведены в табл. 2.

Заключение

В результате проведенных исследований и экспериментов предложен новый способ программного управления яркостью изображения, индицируемого на ЖК экранах бортовых средств отображения информации класса МФЦИ. Основное практическое применение способа относится к ночному режиму полета ЛА, в котором реализуются следующие достоинства способа:

- равномерное изменение яркости изображения на экране МФЦИ в пределах всего индикационного кадра в диапазоне регулировки 0,1...100 % уровня яркости, установленного ручными органами управления;

- инвариантность значений (x, y) -координат цветности к значению параметра управления — масштабного коэффициента k . Сохранение значений координат цветности после внедрения программного способа управления яркостью исключает повторную сертификацию цветовой палитры средства индикации.

Важно заметить, что предложенный программный способ является альтернативным по отношению к способу физического управления яркостью, широко применяемому в современной бытовой технике (мониторы, телевизоры и др.). При этом в большинстве моделей телевизоров с ЖК экранами у пользователя отсутствует возможность снижения яркости свечения экрана "до нуля". Это объясняется неустойчивой работой газоразрядных ламп подсвета в режиме пониженной яркости.

Технические решения, применяемые разработчиками в бытовой технике, обусловлены рядом стандартизованных проектных решений, а именно:

- использование готовых чипов иностранного производства и малогабаритных submodule графических контроллеров (например, VFP зарубежного стандарта PC-104), ориентированных на работу с ЖК матрицами по типовым интерфейсам (LVDS, RGB с сигналами синхронизации, DVI и др.);

- использование Windows-ориентированных таблиц коррекции цветовой палитры в цифро-аналоговых преобразователях кодов RGB, в том числе и гамма-коррекции, в результате которой улучшаются визуальные характеристики восприятия изображения и др.

Однако такие стандартизованные решения оказываются неприменимыми для разработки отечественных изделий авиационной техники, в связи с чем аппаратное управление яркостью имеет ограниченную

область практического применения и может быть заменено предложенным программным способом.

Список литературы

1. **Жаринов И. О., Жаринов О. О.** Бортовые средства отображения информации на плоских жидкокристаллических панелях: учеб. пособие. Информационно-управляющие системы. СПб.: ГУАП, 2005, 144 с.
2. **Костишин М. О., Жаринов И. О., Жаринов О. О.** Исследование визуальных характеристик средств отображения пилотажно-навигационных параметров и геоинформационных данных в авионике // Информационно-управляющие системы. 2014. № 4. С. 61—67.
3. **Руткевич А., Стешенко В., Шишкин Г.** Опыт разработки светодиодных систем отображения графической информации // Chip News. 2007. № 6. С. 36—40.
4. **Соколов В. В., Норка Ю. А., Карпенко С. М., Николаев Д. П.** О визуализации мультиспектральных изображений, сохраняющей локальный контраст // Труды Института системного анализа РАН. 2009. № 45. С. 183—193.
5. **Егоров А. А.** Метод адаптивного контрастирования мультиспектральных изображений для повышения достовер-

ности нечеткой кластеризации // Искусственный интеллект. 2008. № 3. С. 306—314.

6. **Фисенко В. Т., Фисенко Т. Ю.** Компьютерная обработка и распознавание изображений: учеб. пособие. СПб.: Изд-во СПбГУ ИТМО, 2008. 192 с.

7. **Белоусов А. А., Спицын В. Г., Сидоров Д. В.** Применение генетических алгоритмов и вейвлет-преобразований для повышения качества изображений // Известия Томского политехнического университета. 2006. Т. 309, № 7. С. 21—26.

8. **Зотин А. Г., Пахирка А. И., Буряченко В. В.** Алгоритмическое обеспечение экспериментальной системы видеонаблюдения с возможностью улучшения визуального качества видеопотока // Программные продукты, системы и алгоритмы. 2013. № 1. С. 1—18.

9. **Жаринов И. О., Жаринов О. О.** Исследование распределения оценки разрешающей способности преобразования Грассмана в системах кодирования цвета, применяемых в авионике // Программная инженерия. 2014. № 8. С. 40—47.

10. **Жаринов И. О., Жаринов О. О.** Исследование свойства равноконтрастности цветовых пространств, применяемых в авионике // Программная инженерия. 2014. № 11. С. 35—43.

11. **Barber S. et al.** US Patent 7,417,641 B1: Aeronautical chart display apparatus and method; Aug. 26. 2008.

I. O. Zharinov, Associate Professor, Chief of Department, e-mail: igor_rabota@pisem.net, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (University ITMO), Chief of Learning-Scientists Center, SPb Scientific Design Bureau "Electroavtomatika" n. a. P. A. Efimov, **O. O. Zharinov**, Associate Professor, Saint-Petersburg State University of Aerospace Equipment

The Method of Software-Based Brightness Control of Onboard Indication Equipment during Night Flight Operating Mode

The problem of software-based brightness control of onboard indication equipment, designed on LCD panel, during night flight operating mode is considered. New method of image brightness control is proposed. The method is based on using of scaling coefficient and technique of coherent varying of digital codes of basic color components. This method allows to swing image brightness within the range from 0,1 to 100 % of any brightness level manually preset. The main advantage of proposing method is that it allows to keep chromaticity coordinates values for displayed elements on an image within full range of brightness levels, causing no influence on the brightness unevenness for given LCD panel. There are diagrams shown, representing dependance of transmittance coefficient of tested LCD versus scaling coefficient of varying codes for basic colors. Results of brightness level measurements, performed for eight basic colors mainly used in avionics are presented for various values of scaling coefficient.

Keywords: onboard indication equipment, avionics, brightness control, scaling, chromaticity coordinates

References

1. **Zharinov I. O., Zharinov O. O.** *Bortovye sredstva otobrazhenija informacii na ploskih zhidkokristallicheskih paneljah*: ucheb. posobie. Informacionno-upravljajushhie sistemy, SPb.: GUAP, 2005, 144 p. (in Russian).
2. **Kostishin M. O., Zharinov I. O., Zharinov O. O.** *Issledovanie vizual'nyh harakteristik sredstv otobrazhenija pilotazhno-navigacionnyh parametrov i geoinformacionnyh dannyh v avionike. Informacionno-upravljajushhie sistemy*. 2014. N. 4. P. 61—67 (in Russian).
3. **Rutkevich A., Steshenko V., Shishkin G.** *Opyt razrabotki svetodiodnyh sistem otobrazhenija graficheskoi informacii. Chip News*. 2007. N. 6. P. 36—40 (in Russian).
4. **Sokolov V. V., Norka Ju. A., Karpenko S. M., Nikolaev D. P.** *O vizualizacii mul'tispektral'nyh izobrazhenij, sohranjajushhej lokal'nyj contrast. Trudy Instituta sistemnogo analiza RAN*. 2009. N. 45. P. 183—193 (in Russian).
5. **Egorov A. A.** *Metod adaptivnogo kontrastirovanija mul'tispektral'nyh izobrazhenij dlja povyshenija dostovernosti nechetkoj klasterizacii. Iskussvennyj intellect*. 2008. N. 3. P. 306—314 (in Russian).

6. **Fisenko V. T., Fisenko T. Ju.** *Komp'yuternaja obrabotka i raspoznavanie izobrazhenij*: ucheb. posobie. SPb.: Izd-vo SPbGU ITMO, 2008. 192 p. (in Russian)

7. **Belousov A. A., Spicyn V. G., Sidorov D. V.** *Primenenie geneticheskikh algoritmov i veyvlet-preobrazovanij dlja povyshenija kachestva izobrazhenij. Izvestija Tomskogo politehnicheskogo universiteta*. 2006. Vol. 309, N. 7. P. 21—26 (in Russian).

8. **Zotyn A. G., Pahirka A. I., Burjachenko V. V.** *Algoritmicheskoe obespechenie jeksperimental'noj sistemy videonabljudenija s vozmozhnost'ju uluchshenija vizual'nogo kachestva videopotoka. Programmnye produkty, sistemy i algoritmy*. 2013. N. 1. P. 1—18 (in Russian).

9. **Zharinov I. O., Zharinov O. O.** *Issledovanie raspredelenija ocenki razreshajushhej sposobnosti preobrazovanija Grassmana v sistemah kodirovanija cveta, primenjaemyh v avionike. Programmnyaya ingeneria*. 2014. N. 8. P. 40—47 (in Russian).

10. **Zharinov I. O., Zharinov O. O.** *Issledovanie svojstva ravno-kontrastnosti cvetovyh prostranstv, primenjaemyh v avionike. Programmnyaya ingeneria*. 2014. N. 11. P. 35—43 (in Russian).

11. **Barber S. et al.** US Patent 7,417,641 B1: Aeronautical chart display apparatus and method; Aug. 26. 2008.

С. А. Афонин, канд. физ.-мат. наук, вед. науч. сотр., e-mail: serg@msu.ru, НИИ механики МГУ имени М. В. Ломоносова,

К. В. Лунёв, аспирант, e-mail: kirilllunev@gmail.com, МГУ имени М. В. Ломоносова

Выявление тематических направлений в коллекции наборов ключевых слов

Представлены результаты исследований на направлении поиска моделей, алгоритмов и программных средств для определения тематической направленности документа по его ключевым словам. Проведен анализ самих ключевых слов, а именно изучена степень их абстрактности. Методы, использованные в работе, основаны на теоретико-графовых алгоритмах.

Ключевые слова: тематическое моделирование, обработка естественного языка, алгоритмы на графах

Введение

Во многих информационных системах с объектами различных типов, например, текстовыми документами, аудио- и видеоматериалами, файлами электронных таблиц, могут быть связаны наборы ключевых слов или тегов. Ключевое слово (также известное как тег, ярлык, метка, дескриптор) представляет собой идентификатор, содержащий в себе знания об информационном ресурсе. Область применения тегов очень обширна. Ключевые слова могут использоваться в базах научных статей и социальных сетях, в информационных ресурсах и блоговых системах. Такими метками можно описывать текстовые документы (новости, сообщения электронной почты, публикации, посты), видео- и аудиофайлы, электронные таблицы, отдельные файлы и другие объекты.

Знание ключевых слов позволяет получить достаточно много информации о документе, которому они принадлежат. Например, по ключевым словам можно определить тематику всего документа. Однако это не единственное из возможных применений для тегов. Анализ наборов ключевых слов позволяет пользователю составить представление о коллекции документов в целом. Примером такого анализа является облако тегов. Облако тегов — это визуальное представление ресурса некоторым набором ключевых слов. Размеры и цвет шрифта отдельного тега определяют его важность на данном ресурсе. Формируется такое облако по результатам анализа документов, входящих в ресурс, и по наборам ключевых слов, соответствующих этим документам.

Изучение ключевых слов помогает в решении задач поиска документов в информационных системах. В корпоративных системах или базах научных статей распространенным методом поиска является поиск по ключевым словам. Этот метод особенно уместен, если документы вообще не представляют текстовой информации. Примерами таких документов служат изображения, аудио- и видеозаписи. В этом случае

поиск значительно затрудняется и, если документы не помечены ключевыми словами, приходится применять алгоритмы, анализирующие контент, что зачастую негативно отражается на скорости и качестве поиска. В рамках задачи информационного поиска возникает также необходимость проведения ассоциативного поиска — поиска, при котором запрос содержит ключевые слова, близкие по смыслу меткам документов, но не совпадающие с ними в точности. В дополнение к представленным выше соображениям отметим, что с использованием анализа тегов пользователю информационной системы могут в автоматическом режиме добавляться документы, которые, возможно, будут ему интересны.

Другим важным применением тегов является автоматическая классификация документов — одна из базовых задач информационного поиска, которая заключается в отнесении документа к одной из нескольких категорий на основании анализа содержания информации о документе. В рассматриваемом случае в качестве такой информации используются именно ключевые слова.

Еще одно направление, в котором можно эффективно использовать анализ ключевых слов, — это обработка естественного языка. Предложения и тексты могут быть представлены как наборы ключевых слов. После этого можно проводить анализ текстов, используя методы работы с наборами ключевых слов. Такой подход применяется в задачах определения смысловой близости предложений и тестов, кластеризации и обобщения текстовых документов.

В рамках работы, результаты которой представлены в настоящей статье, рассматривается задача определения тематической направленности документа по его ключевым словам. Необходимо для каждого документа выделить одно или несколько ключевых слов из всего корпуса документов так, чтобы эти слова указывали на тематику всего документа. Проводится также анализ самих ключевых слов, а именно изучается степень абстрактности их значений. Словами с высокой

степенью абстрактности являются слова более общего характера, они обладают более широким значением. Зачастую, слово с большим уровнем абстрактности является гиперонимом для некоторых слов с меньшей степенью абстрактности. Понимание термина "абстрактность слова" необходимо для составления иерархических отношений между словами. Методы, использованные в работе, основаны на теоретико-графовых алгоритмах.

Документ представляется в виде множества ключевых слов, ассоциированного с этим документом. По заданной коллекции документов строится граф ключевых слов. Вершины этого графа соответствуют ключевым словам, а взвешенные ребра отражают факт вхождения слов в один набор. Для изучения множества документов используются алгоритмы на графах (вычисление кратчайших путей, различных количественных характеристик вершин графа) и статистический анализ. С помощью графа выявляются тематические зависимости между документами, а также тематические особенности того или иного документа. Для решения поставленной задачи используется характеристика вершин графа — центральность. В работе [1] представлены различные способы вычисления абстрактности тега, основанные на алгоритмах на графах и семантических сетях. В работе [2] степень абстрактности ключевого слова определяется с помощью статистических методов. Для выявления тематических зависимостей авторы работ [3, 4] использовали вероятностные модели и методы машинного обучения. В работах [5, 6] изучены различные меры близости ключевых слов и алгоритмы поиска в "фолксономиях" — системах, в которых пользователям позволяет интерактивно помечать документы метками (тегами).

Трудность решения поставленной задачи заключается в том, что о данном ключевом слове нет никакой информации, кроме его причастности к некоторому набору. Этот факт означает, что не используются никакие семантические знания о слове. В качестве тестовых данных используется набор ключевых слов научных публикаций в области как естественных, так и гуманитарных наук. Отметим, что дополнительную сложность создает наличие многозначных ключевых слов и аббревиатур.

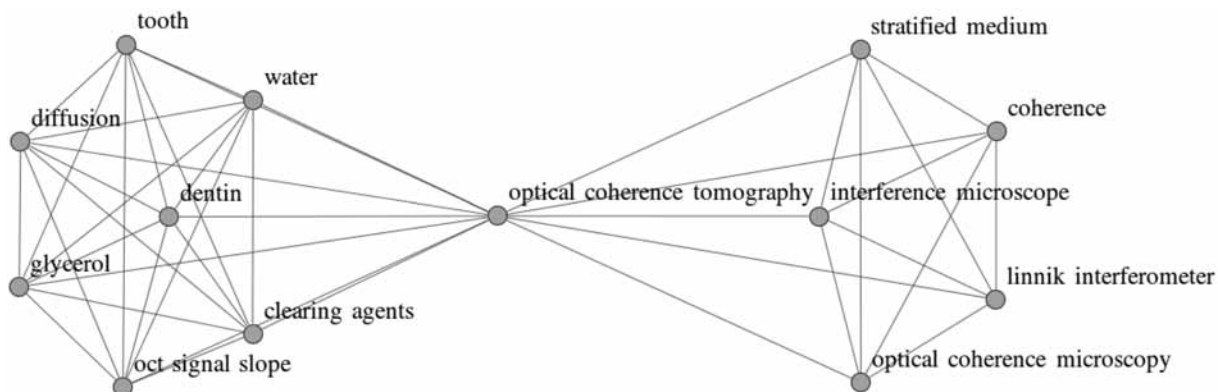


Рис. 1. Пример небольшой компоненты связности графа тегов

1. Постановка задачи

Дано множество документов D и множество ключевых слов W . Каждый элемент $d_i \in D$ представлен набором из k_i ключевых слов из множества W : $d_i = (w_{i,1}, w_{i,2}, \dots, w_{i,k_i})$. Необходимо разработать методы определения тематики для каждого документа коллекции. Под тематикой следует понимать название некоторой области знаний, дисциплины, специальности или направления. Определив такое слово для документа, можно с высокой вероятностью сказать, о чем этот документ. Тематический тег — это такой тег из W , который может являться тематикой документа. Таким образом, необходимо из множества ключевых слов W выделить подмножество $T \subset W$ тематических тегов и для каждого документа из коллекции определить соответствующее ему множество тематических тегов, т. е. определить отображение $f: D \rightarrow 2^T$. Отметим, что принадлежность тега к тематике зависит от набора документов.

Для решения поставленной задачи в следующих разделах используется понятие абстрактности ключевого слова. Под абстрактностью понимается степень общности значения слова. Необходимо разработать такую меру абстрактности $A: W \rightarrow \mathbb{R}^+$ (\mathbb{R} — множество действительных чисел), чтобы большим значениям меры соответствовали слова более широкого значения, а меньшим значениям — слова более конкретно, обладающего определенной спецификой значения.

2. Граф ключевых слов

В настоящем разделе представлен граф ключевых слов, который понадобится в следующих разделах для описания алгоритмов определения абстрактности ключевого слова, смысловой близости пары ключевых слов, а также определения тематической направленности набора таких слов.

В вершинах рассматриваемого графа стоят ключевые слова, а ребра отражают тот факт, что пара соответствующих ключевых слов входит в некоторый набор. Более формально граф тегов — ненаправленный граф $G = (W, E, c)$, в котором множество вершин есть множество ключевых слов, а множество ребер — подмножество $E \subseteq W \times W$. Пара тегов $w_1,$



Рис. 2. Пример подграфа графа тегов

w_2 соединена ребром, если существует хотя бы один документ $d \in D$, такой, что $w_1, w_2 \in d$. $c: E \rightarrow \mathbb{R}^+$ обозначает вес ребра и определяется как число документов, в которое входит данная пара тегов, т. е. $c(t_1, t_2) = \text{card}\{d \in D | t_1, t_2 \in d\}$.

На рис. 1 представлен такой граф. Он состоит из небольших полных подграфов, соединенных друг с другом. Появление полных подграфов обусловлено тем, что теги, входящие в один набор, будут иметь связи друг с другом. Часть графа большего размера представлена на рис. 2.

3. Определения абстрактности слова

В настоящем разделе представлены алгоритмы определения степени абстрактности ключевых слов на основе графа ключевых слов. В наборах тегов присутствуют слова, которые указывают на общую тематическую направленность документа, а также слова, отражающие более узкую специализацию (термины). Проиллюстрируем это на примере следующего набора тегов (подчеркнуты слова более общего значения):

[рынок труда, профессиональная ориентация, прогнозирование, модель, алгоритм]

Трудность решения поставленной задачи обусловлена тем обстоятельством, что необходимо уметь отделять действительно абстрактные по значению слова от слов популярных и, как следствие, часто используемых в наборах ключевых слов.

3.1. Определение степени абстрактности на основе свойства центральности

В теории графов существует характеристика важности вершины графа, которая по-русски именуется *центральность*. Для того чтобы определить влияние вершины внутри графа, существует несколько различных мер. Основные из них перечислены далее.

• Центральность по посредничеству (*betweenness centrality*) — мера, которая определяется через число кратчайших путей, проходящих между всеми парами вершин в графе через данную вершину: $bc(i) = \sum_{s,t \in V \wedge s \neq i \wedge t \neq i} \frac{n_{s,t}^i}{n_{s,t}}$, где $n_{s,t}$ — число кратчайших путей через вершины s и t ; $n_{s,t}^i$ — число кратчайших путей

через вершины s и t , проходящих через i ; V — множество вершин графа.

• Центральность по близости (*closeness centrality*) — мера, основанная на средней длине кратчайшего пути между исходной вершиной и всеми другими

вершинами графа: $c(i) = \frac{\left[\sum_{j \in V} \text{dist}(i, j) \right]^{-1}}{k}$, где $\text{dist}(i, j)$ —

длина кратчайшего пути между вершинами i и j ; k — число вершин из компоненты связности вершины i ; V_i — множество вершин из этой компоненты связности.

• Центральность по степени (*degree centrality*) — мера, в которой важность вершины равна её степени (числу инцидентных ребер).

• Центральность собственного вектора (*eigen vector centrality*) — мера, описанная в работе [7], она вычисляется по следующей формуле: $x(i) = \frac{1}{\lambda} \sum_{j \in V} a(i, j) x(j)$, где

a — матрица смежности графа, λ — константа. Это выражение может быть переписано в векторной форме следующим образом: $Ax = \lambda x$. Большее значение меры ставится той вершине, которая соединена с большим числом вершин, меры которых высоки.

• PageRank-центральность (*PageRank centrality*) — алгоритм, используемый Google для ранжирования страниц (метод определения важности или популярности документов) [8]. Согласно этому алгоритму веб-страница имеет больший вес, если на нее много ссылок из других веб-страниц, также имеющих большой вес. Заметим, что в данном виде алгоритм применим не только для веб-страниц, но и для любых графов. Общая идея этого алгоритма близка к идее, которая реализуется алгоритмом вычисления *eigen vector centrality*. Однако в уравнениях, которые используются алгоритмом PageRank, вместо собственных значений присутствует коэффициент затухания q . Физический смысл этого коэффициента в том, что пользователь имеет некоторую вероятность перехода с одной веб-страницы на другую по ссылке. При этом он может никуда не переходить и закончить случайное блуждание по сети. Таким образом, q — это вероятность того, что пользователь перейдет по ссылке. Получаемое уравнение имеет следующий вид:

$$\text{PageRank}(i) = (1 - q) + q \sum_{j=1}^n \frac{\text{PageRank}(j)}{C_j},$$

где C_j — число ссылок в документе j . В программной реализации параметр q определен как 0,9.

Далее используется идея, что слишком длинный путь между вершинами в графе обычно не означает хотя бы какой-то связи между словами. По этой причине программная реализация алгоритмов *Betweenness Centrality* и *Closeness Centrality* сделана таким образом, чтобы длинные пути не учитывались. Для этого соответствующие меры для вершины считаются не во всем графе, а в подграфе соседей на расстоянии 3 от вершины. Для каждого из алгоритмов за меру абстрактности тега принимается мера центральности соответствующей вершины. Результаты рабо-

ты программной реализации алгоритмов представлены в приложении А. Отметим, что под степенью абстрактности следует понимать общность значения слова именно в рамках заданного корпуса слов. Это, например, означает, что слово "вероятность" можно считать абстрактным потому, что можно предположить, что существует много статей, посвященных некоторым проблемам из теории вероятности. По этой причине существует много различных слов-терминов, имеющих прямое отношение к слову "вероятность".

Резюмируя представленные выше соображения, необходимо отметить, что степень абстрактности слова является достаточно субъективной величиной. По этой причине проведение объективной количественной оценки результатов работы алгоритмов на больших объемах данных затруднительно. В качестве основного метода проверки адекватности предлагаемого метода используется выборочная экспертная оценка результатов.

3.2. Окончательный алгоритм

В качестве окончательного алгоритма авторами принят алгоритм, смешивающий результаты представленных выше алгоритмов, а именно:

- для каждого алгоритма вычисляется вектор из центральных вершин;
- каждый вектор нормируется;
- сумма векторов является вектором, на i -й координате которого стоит степень абстрактности i -го ключевого слова. Последняя сумма нормируется и выводится в качестве ответа.

4. Определение тематической направленности набора ключевых слов

В множестве ключевых слов, которые приписаны документам из некоторой коллекции, можно выделить "тематические теги" — ключевые слова, семантическое значение которых является названием некоторого "широкого" направления. Примерами таких тегов могут быть следующие слова: динамика, статистика, право, численные методы. Выделение таких слов из набора является простой задачей для человека, но очень сложной для машины. В настоящем разделе описан алгоритм, определяющий тематику набора тегов, а также приведены результаты его работы на реальных данных. Предложенные алгоритмы опираются на понятие абстрактности ключевого слова, а также на алгоритм ее определения.

4.1. Описание алгоритма определения тематических тегов

Если для каждого слова из коллекции определить степень его абстрактности, то можно все множество слов разделить на следующие три типа:

- абстрактные теги;
- тематические теги;
- термины.

Абстрактные теги, такие как **моделирование, структура, эффективность, анализ**, имеют наибольшие значения абстрактности и являются самыми общими по смыслу. По таким тегам нельзя определить, о чем документ. Вместе с тем существует

класс терминов — ключевых слов с наименьшим уровнем абстрактности. Человек может легко определить тематику документа, если знает значения таких слов. Примерами таких слов могут быть следующие: **дисульфид молибдена, параметры межпланетной среды, дипептиды, пьезоакселерометрия** и др. Между ними лежат тематические теги. Трудность решения поставленной задачи заключается в том, чтобы определить границы, где заканчивается один класс и начинается другой. Подчеркнем, что выбор этих границ субъективен: тег **динамика ударных волн** является, с одной стороны, названием некоторой области научного знания, а с другой стороны, по объективным причинам складывается представление, что это ключевое слово имеет достаточно узкоспециальный характер. Помечать такой тег тематическим или нет, зависит от конкретной решаемой задачи. Очевидным является тот факт, что подавляющее большинство тегов должно являться терминами, а меньше всего должны быть абстрактными ключевыми словами. При этом термины, в отличие от абстрактных слов, редко повторяются. Для определения степени абстрактности слова будем использовать алгоритм, описанный в разд. 3. Для решения поставленной задачи рассмотрим типичные наборы ключевых слов. Далее приведены примеры наборов, содержащих тематический тег (предполагаемый тег подчеркнут).

[экологические процессы, социальная история, природные ресурсы]

[беспризорность, политика государства, детский дом, интеграция, адаптация]

[экономика, модернизация экономики, подготовка кадров, система повышения квалификации]

[динамика, удар, механические свойства, замедление, пьезоакселерометрия]

[удар, динамика, летательный аппарат, экран, модель, физическое моделирование]

Следует заметить, что очень часто наборы представляются без тематических тегов. Можно предположить, что авторы соответствующих статей считают такие ключевые слова бесполезными для определения их содержания. Если человек хочет прочитать статью о схемах Рунге-Кутты, то он должен и без явного указания понимать, что речь идет о численных методах. Первым шагом алгоритм проходит по всем имеющимся наборам и выбирает из каждого набора тег, обладающий самой большой степенью абстрактности. Среди них, предположительно, присутствуют практически все абстрактные и тематические теги и некоторая часть тегов-терминов. На рис. 3 изображены отсортированные значения абстрактности полученных тегов. По оси абсцисс отложены порядковые номера тегов. Видна экспоненциальная зависимость: существует большое число тегов с относительно малым значением степени абстрактности, а тегов с большим значением немного.

После этого значения степеней абстрактности логарифмируются. График для значений логарифмов степеней абстрактностей приведен на рис. 4. По оси

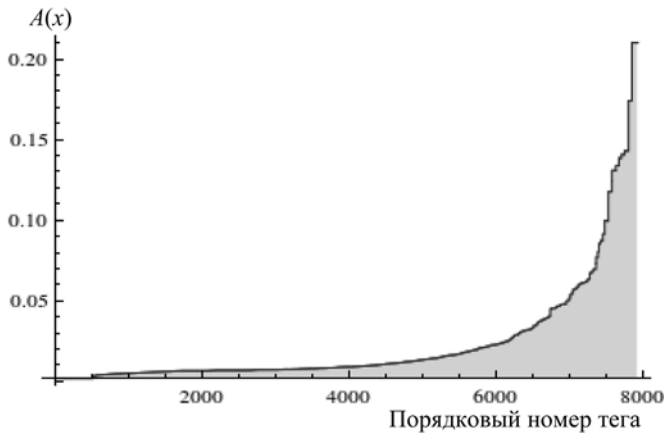


Рис. 3. Значения степени абстрактности

абсцисс также отложены порядковые номера тегов. Этот шаг сделан для того, чтобы "выпрямить" экспоненту. Можно заметить, что график гораздо больше стал походить на прямую. На последнем шаге алгоритма рассчитывается среднее значение, от которого берется экспонента, что позволяет вернуться к реальным значениям степеней абстрактности.

Обосновать выбранный алгоритм можно следующим образом: если предположить, что в среднем в каждом наборе присутствует один тематический тег (некоторые из которых могут быть достаточно конкретными значениями и граничить по уровню абстрактности с тегами-терминами), то выбор необходимой области значений степеней абстрактности должен зависеть целиком от этого подмножества тегов. После этого естественно взять некоторое среднее значение, назначить центром, рядом с которым будет максимальное число искомых тематических тегов. Однако поскольку значения степеней абстрактности имеют экспоненциальный график, а число тегов, близких к терминам, очень высоко, то их среднее значение создает перевес в сторону тегов-терминов. По этой причине, чтобы уменьшить их влияние, от всех значений абстрактностей берется логарифм.

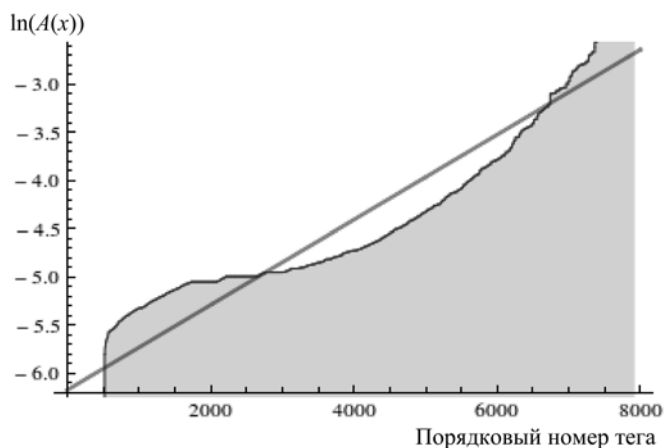


Рис. 4. Значения логарифмов степени абстрактности

Формулу, определяющую оптимальное значение степени абстрактности p , в окрестность которого попадет максимальное число искомых тематических тегов, можно представить в следующем виде:

$$p = \exp M \left(\left\{ \log x \mid x = \max_y (A(y) \mid y \in P), p \in P \right\} \right),$$

где P — множество всех наборов; $A(y)$ — уровень абстрактности тега y ; $M(X)$ — среднее значение числа множества X .

После определения параметра p остается выбрать интервал, содержащий значение p . Например, можно отступить в обе стороны от p на ε и обозначить множество тегов, таких что $A(x) \in [p - \varepsilon, p + \varepsilon]$, множеством тематических тегов.

4.2. Описание алгоритма выбора тематики набора

После того как сформировано множество тематических тегов, появляется возможность определить тематику целого набора. Для этого необходимо взять тег с наибольшим уровнем абстрактности. Если такой тег лежит в множестве тематических, то можно выдать его в качестве ответа. В противном случае необходимо найти ближайший тематический тег из графа тегов. Для этого сначала берутся соседи исходной вершины. Полученное множество просматривается на наличие в нем тематических тегов. Если такие теги присутствуют, то все они возвращаются в качестве ответа (таким образом, для набора выбирается несколько возможных тем), иначе просматриваются соседи соседей и т. д. В некоторых случаях определить тематическую направленность не представляется возможным. Очевидно, это происходит тогда, когда в компоненте связности самого абстрактного тега в наборе не присутствует ни одного тематического тега.

Отметим, что учитывается тот факт, что если расстояние между парой тегов в графе велико, то нельзя выделить какую-либо семантическую связь между этими тегами. Также заметим, что если радиус круга соседей будет слишком велик, то очень вероятно, что в множество тематических тегов попадет слишком много слов. Для разрешения этих трудностей введен параметр r , который показывает максимальное возможное расстояние в графе между исходным тегом и ближайшим найденным тематическим. Если в радиусе r тематический тег найти не удалось, то тема набора не определена. В программной реализации алгоритма значение r равно 2. Алгоритм, соответствующий представленным выше соображениям, был реализован на языке системы *Mathematica*.

5. Тестовые испытания

В настоящем разделе представлены результаты тестовых испытаний программной реализации алгоритмов определения степени абстрактности ключевого слова, тематических тегов и тематики документа, а также описаны методы предварительной обработки исходных данных и исправления ошибок. В качестве таких тестовых данных в работе использован корпус ключевых слов научных публикаций.

Коллекция представляла собой вручную составленные списки ключевых слов для публикаций технического и гуманитарного профиля, полученные из различных источников, включая сеть Интернет. По этой причине в этих данных присутствовали полученные при их формировании ошибки и неточности.

5.1. Подготовка и анализ тестовых данных Предобработка данных

Предобработка данных — необходимая мера для увеличения точности работы алгоритмов. По причине того, что данные для тестовых испытаний вводились вручную, были допущены ошибки, среди которых:

- неправильный разделитель (или отсутствие разделителя) между ключевыми словами;
- неправильное число символов пробелов между словами;
- дублирование тегов;
- опечатки в словах;
- ключевые слова в верхнем регистре.

Присутствовали также и другие трудности в обработке, а именно использование:

- дефисов для обозначения переноса строки;
- аббревиатур;
- различных форм слов.

Несмотря на то что ошибки присутствовали в меньшей части коллекции, они негативно влияли на результаты работы алгоритмов. Поскольку в работе применялись статистические методы и алгоритмы на графах, то необходимо было добиться того, чтобы число слов, обозначающих одно и то же понятие, было минимально.

Из всех представленных выше трудностей не удалось избавиться в автоматическом режиме от опечаток и использования разных форм слов. К мерам, которые применялись для улучшения качества тестовых данных, относятся следующие:

- все ключевые слова переводились в нижний регистр;
- самые популярные из аббревиатур вручную сопоставлялись со своими развернутыми формами;
- использовалось несколько разделительных символов;
- длинные строки без разделителей разделялись по символам пробелов;
- в ключевом слове убирался дефис, если уже существует такое же слово без дефиса.

При этом заметим, что длинные строки без разделителей в действительности могут представлять собой единственный тег:

[оценка экономического косвенного эффекта от проекта информатизации]

Набор тегов, разделенных по пробелу:

[шахта метан утилизация газогенераторная станция]

После реализации предобработки данных число ключевых слов уменьшилось на 14 %.

Анализ тестовых данных

Всего в тестовых данных присутствовало 8932 набора, которые в сумме дали 17 428 ключевых слов.

На рис. 5 представлена гистограмма числа слов в наборах. Видно, что в среднем люди приписывают публикациям 3—4 тега. При этом "хвост" справа достаточно тяжелый и наборы длин от 4 до 10 встречаются достаточно часто. На рис. 6 показано число упоминаний тега в корпусе. Можно заметить, что даже после выполненной предобработки большая часть тегов встречалась только один раз. Хотя первичная обработка не исправила все ошибки в данных, ясно, что подобное распределение получится и при идеально отобранных данных. Теги, использованные единожды, показывают узкую специализацию тематики публикации. Очевидно, что таких узких направлений очень большое число, поэтому подобные результаты будут справедливы и для любых наборов данных, где теги задаются произвольно, а не из предварительно фиксированного множества слов. В среднем каждый тег употребляется 1,69 раза. Список самых популярных тегов выглядит следующим образом:

наночастицы, инновации, метод конечных элементов, механические свойства, динамика, наноструктуры, прочность, научный потенциал, удар, структура, остаточные напряжения, компьютерное моделирование, управление, модель, оптимизация, мониторинг, образование, математическое моделирование, математическая модель, моделирование

Популярность некоторых из этих слов является следствием высокой степени абстрактности (**моделирование**), но некоторые из них попали в список, потому что некоторая тема может быть популярной (по крайней мере, в рамках данной коллекции). По этой причине теги, относящиеся к этой тематике, часто могут быть использованы в списке ключевых слов (**наноструктуры, метод конечных элементов**). Способы отделения друг от друга тегов этих двух видов рассмотрены в разд. 3. Однако уже сейчас видно, что наивный алгоритм подсчета числа вхождений тега в коллекцию не дает необходимого результата.

На рис. 7 показано число наборов, обладающих соответствующей долей неуникальных тегов (т. е. тех тегов, которые упоминаются хотя бы в двух наборах из коллекции). Другими словами, для каждого

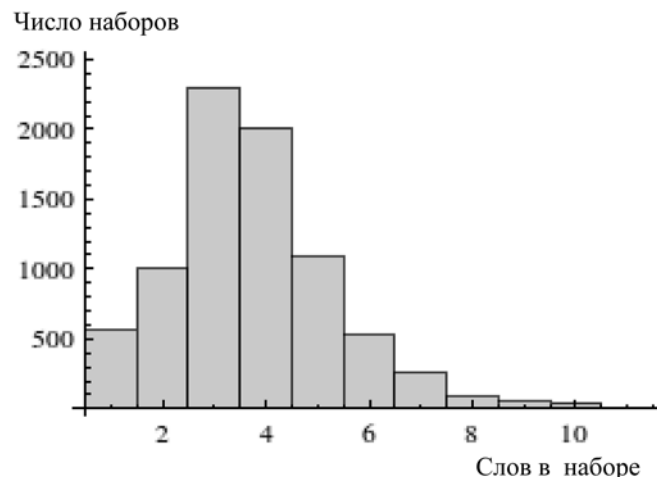


Рис. 5. Распределение длин наборов

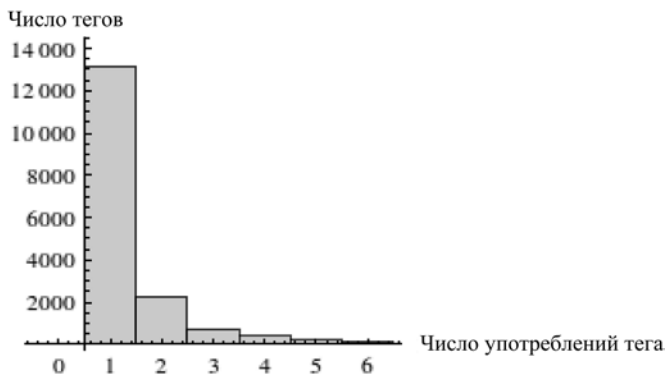


Рис. 6. Число упоминаний тегов в корпусе

набора подсчитан процент неуникальных ключевых слов и по этим данным построена гистограмма. Важно отметить, что именно неуникальные теги дают возможность дальнейшего анализа. Если бы наборы в основном состояли из уникальных тегов, то использование графов и статистического анализа не привело бы к достижению каких-либо результатов.

Самое популярное значение доли неуникальных тегов — 1. Этот факт означает, что теги всех наборов этой категории не являются уникальными, несмотря на то что коллекция состоит по большей части из уникальных тегов. Пример такого набора:

[образование, наука, высшая школа, идеология, математика, филология, история, педагогика, биология]

Зачастую теги наборов такого типа являются очень абстрактными понятиями, что в редких случаях позволяет понять тематику документа. Второе по популярности значение — 0. Оно показывает, что существует много наборов, целиком состоящих из уникальных тегов. Причины возникновения таких наборов состоят в следующем:

- ключевые слова являются слишком узкоспециализированными, например, **фемтосекундная спектроскопия, лазерные солитоны, дипирролилметаны, подповерхностный радиолокатор**;
- ключевые слова представлены на другом языке — **control, sensitivity, equilibria, chaos**;
- ключевые слова имеют слишком большую длину, например, **гетеродинный спектрометр высокого разрешения**.

Следует, однако, отметить, что основная причина именно в узкой специализации тегов.

Между значениями 0 и 1 находится более половины всех наборов. Средняя доля неуникальных тегов по всей коллекции равна 0,53, т. е. в среднем половина тегов набора встречается в некотором другом объекте коллекции. Типичный набор состоит из нескольких абстрактных тегов, указывающих на общее направление работы и дисциплины, и нескольких тегов, позволяющих понять, о чем конкретно представлен документ.

Исходя из перечисленных выше факторов, представляется возможным изучать методы автоматического определения тематики документа и особенности алгоритмов ассоциативного поиска. При этом

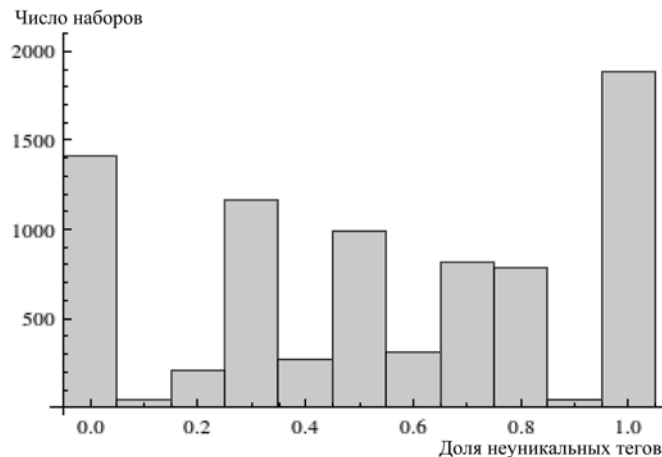


Рис. 7. Число наборов с соответствующей долей неуникальных тегов

логичным инструментарием для решения поставленных задач являются графы, которые были введены в предыдущем разделе. Такие графы будут иметь достаточную связность для дальнейшего анализа и возможности применения алгоритмов, которые представлены в предыдущих разделах.

Для графа тегов, построенного по данным, были определены компоненты связности и вычислены их размеры. Общее число компонент связности — 1856, что, очевидно, очень много для графа из 17 428 вершин. Однако, как на рис. 8 показывает график зависимости номера компоненты и ее размера (ось ординат логарифмическая), наибольшая компонента связности содержит более половины всех тегов (11 558), а вторая по величине имеет лишь 92 вершины. Начиная с 38-й позиции в компонентах содержится менее 10 вершин.

5.2. Дополнительный тестовый набор данных

По причине того что имеющийся набор данных недостаточно велик, был разработан алгоритм сбора информации о ключевых словах научных статей из сети Интернет. Для этого был использован интерфейс программирования приложений (API) поисковой системы Яндекс, представляющей функциональные возможности получения поисковой выдачи по запросу. Суть алгоритма в следующем. На вход алгоритма подается одно ключевое слово <KEYWORD>. В поисковую систему отправляется запрос вида "mime:pdf keywords: /5 <KEYWORD>". Этот запрос означает, что необходимо найти документы формата pdf, в которых после слова "keywords:" на расстоянии не более 5 слов находится заданное слово <KEYWORD>. Поисковая система возвращает фрагменты релевантных документов (сниппеты). Ожидается, что значительная часть фрагментов будет содержать списки ключевых слов некоторых научных статей. Из найденных фрагментов выделяются кортежи ключевых слов. Собранные ключевые слова добавляются в множество ключевых слов. Тег-запрос добавляется в список использованных ключевых слов. Новое ключевое слово <KEYWORD> выбирается из разности множества всех ключевых слов и множества использованных. Действия алго-

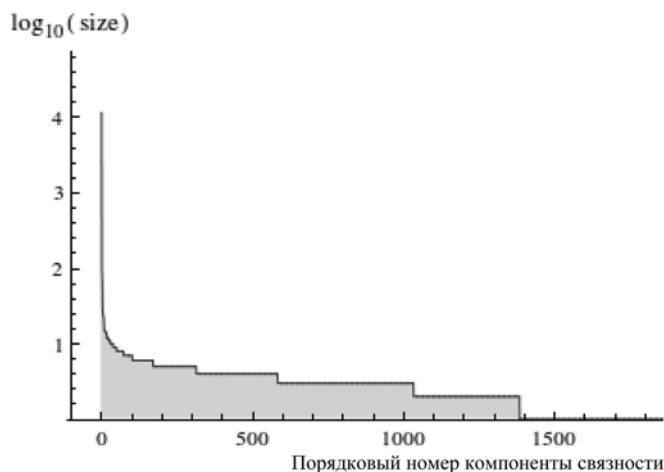


Рис. 8. Размеры компонент связности

ритма повторяются до тех пор, пока не будет собрана база ключевых слов достаточного размера.

Недостатком программной реализации представленного алгоритма является то обстоятельство, что внешняя поисковая система ограничивает число запросов в день. По этой причине сбор необходимых данных занимает продолжительное время. Чтобы ускорить этот процесс, для каждого запроса загружается максимально возможное число документов. Как следствие, ключевые слова, по которым строился запрос, встречаются гораздо чаще в собранном множестве ключевых слов. Такой "перекосяк" негативно влияет на статистические параметры выборки и ухудшает качество работы реализаций алгоритмов. Другой недостаток состоит в том, что если брать слишком большое число документов, то "хвост" выдачи становится менее релевантен и в выборку добавляются "мусорные" данные.

Тем не менее алгоритм решает важную задачу — восполняет недостаток данных. С помощью программной реализации было собрано более 380 000 наборов ключевых слов. Для них были проведены методы предобработки данных, описанные в предыдущем подразделе. Дополнительно к этому удалялись наборы без разделителей. Вероятнее всего, такие наборы — это обычные предложения со словом keywords. Также были удалены наборы, в которых ключевые слова имеют слишком малую длину (обычно в такие сниппеты попадали инициалы авторов статей). Далее это множество данных обозначается как *данные из Веб*, а первая коллекция именуется *чистыми данными*.

5.3. Результаты тестовых испытаний

Определение уровня абстрактности ключевого слова

Ниже приведены самые абстрактные слова, полученные при использовании программной реализации описанного выше алгоритма на чистых данных.

моделирование, модель, образование, оптимизация, управление, структура, математическая модель, математическое моделирование, мониторинг, прогнозирование, инновации, эффективность, методика, личность,

прочность, эксперимент, оценка, история, методы, развитие, анализ, здоровье, инновационная деятельность, культура, качество, свойства, модернизация, синтез, надежность, самоорганизация, адаптация, конкурентоспособность, интеграция, студенты, безопасность, компетенции, взаимодействие, технологии, диагностика, наука, государство, компьютерное моделирование, инновационное развитие, устойчивость, компетентностный подход, динамика, технология, высшая школа, наночастицы, метод конечных элементов

В целом получены неплохие результаты по экспертной оценке. Из выделенных слов значительную часть можно назвать абстрактными в некотором смысле. Смешивание помогло избавиться от явных выбросов в каждом из алгоритмов и несколько усреднить результат. Тем не менее добиться заметного повышения качества не удалось, поскольку представленные алгоритмы имеют одну природу и решают схожие задачи. По этой причине они зачастую ошибаются на некоторых данных одновременно, что влечет за собой ошибку в результатах работы общего алгоритма.

Далее приведены результаты работы программной реализации алгоритма на данных из Веб.

development, data mining, environment, evaluation, model, management, machine learning, modelling, growth, reliability, neural networks, design, stability, learning, security, uncertainty, clustering, education, performance, modeling, optimization, simulation

Для этого набора получены схожие по качеству результату. Однако можно заметить, что некоторые слова продвигаются вверх в силу того, что они были использованы в запросах. Таким словом, например, является термин **machine learning**, который не должен был войти в множество абстрактных слов.

Результаты тестовых испытаний определения тематических тегов

Значение искомого параметра p в описанном ранее алгоритме оказалось равным 0,0122. На чистых данных для удобства тестирования был выбран интервал $[0,012, 0,013]$. Теги, которые имеют такую степень абстрактности, определены как тематические. Таких ключевых слов обнаружено 65. В приложении Б приводится полный список определенных тематических тегов. Таким образом, алгоритм определил 65 тегов, из которых 19 при любых обстоятельствах являются тематическими потому, что это название дисциплин и направлений науки. Еще 13 тегов субъективно можно считать тематическими. Таким образом, точность результата составила 49,2 %.

Проверка полноты результатов в полной мере не представляется возможной потому, что четких этапов тематического тега не существует, а проверить справедливость причисления около 18 000 тегов к тематическим практически невозможно. Вместо этого авторами были рассмотрены соседние интервалы степеней абстрактности и отобраны те теги, которые обязаны быть в списке тематических. В приложении Б представлены списки истинных тематических тегов среди других интервалов степеней абстрактности. На рис. 9 изображена гистограмма числа реальных

тематических тегов в других интервалах степеней абстрактности. На рис. 9 видно, что наибольшее значение получилось именно в интервале, в который попало значение p , определенное алгоритмом.

Несмотря на то что параметр p был определен практически точно, очень велико число тегов, которые не должны были быть помечены абстрактными. Причиной этого может быть неточность работы алгоритма определения абстрактностей, а также сама природа данных. Результаты работы программной реализации алгоритма на данных из Веб приведены также в приложении Б. Точность составила 34,3 %, что значительно меньше, чем при работе с чистыми данными. Причиной этого может служить качество алгоритма сбора данных.

Результаты тестовых испытаний определения тематики набора

Некоторые результаты работы программной реализации алгоритма определения тематической направленности набора представлены в таблице.

В примере 1 показано верное определение тематики. При этом можно заметить отсутствие тематического тега в исходном наборе. Примеры 9, 10 демонстрируют случаи, когда тематический тег не определен или определено слишком большое число таких тегов.

В целом следует отметить, что алгоритм допускает достаточно много ошибок. Очень часто ошибки возникают на наборах, состоящих только из редких тегов-терминов. Причина в том, что теги в подобных наборах не соединены напрямую с тематическими, а увеличение пути сильно уменьшает качество результатов. Не определяется тематика и у тех наборов, теги которых попали в малые компоненты связности без тематических тегов. Однако основная трудность — это накопление ошибок разных алгоритмов. Сначала тег ошибочно получил высокий уровень абстрактности, затем попал в список тематических и в итоге неправильно определил тематику набора. В случае с данными из Веб ошибка накапливается с этапа сбора данных, поэтому результаты становятся неприемлемо плохими.

Несмотря на отмеченные недостатки, предложенный подход потенциально может быть улучшен. Для этого можно использовать все теги из набора для опре-

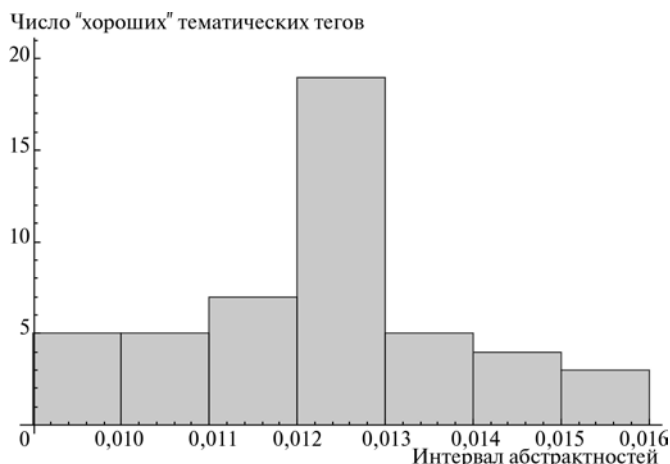


Рис. 9. Число тематических тегов на каждом интервале

Наборы и предсказанные им тематические направления

№ примера	Набор ключевых слов	Возможная тематика набора
1	Выпуклое программирование, принцип лагранжа, теорема Куна-Таккера в недифференциальной форме, параметрическая задача, минимизирующая последовательность, двойственность, регуляризация	Оптимальное управление
2	Состояние метакультуры, культура среды, границы, этика творчества, личность	Биометрия, массовая культура, педагогическая деятельность
3	Окружающая среда, биосферная совместимость, система жизнеобеспечения	Гидродинамика, факторный анализ
4	Персонализированное обучение, подготовка учителя информатики, синергетический подход	Конструирование, педагогическая деятельность
5	Жидкие кристаллы, ориентирующие слои, аморфный гидрогенизированный углерод	Жидкие кристаллы
6	Прогноз, формы предвидения, интерпретация темпоральных модусов, аксиология, социальное противоречие	Конструирование, массовая культура
7	Металлические материалы, термическая обработка, структура, свойства	Конструирование, физическое моделирование
8	Инклюзия, инвалидность, сопровождение, студенты-инвалиды	Массовая культура
9	Модуль выпуклости, равномерно выпуклая функция, равномерно выпуклое множество	—
10	Урбанизированная территория, гуманитарный баланс, корреляция, регрессия, математическая модель	Биометрия, кинематика, конструирование, нелинейные колебания, физическое моделирование

деления тематики, проверять абстрактности вершин на пути от данного тега к тематическому и использовать многие другие эвристики. Это предмет исследования на ближайшее будущее. Однако уже сейчас можно констатировать, что алгоритм способен с некоторой точностью решать поставленную задачу.

Заключение

В статье введено понятие тематического тега и предложены алгоритмы определения степени абстрактности ключевых слов и выявления тематических тегов в коллекции наборов ключевых слов. Предлагаемые алгоритмы основаны на представлении коллекции наборов ключевых слов в виде нескольких ориентированных графов и вычислении ряда их ха-

рактических. Все предложенные алгоритмы были реализованы на языке *Mathematica*, и в работе приведены результаты тестирования на реальных массивах данных. Отмечено, что масштабное тестирование сильно затруднено потому, что не существует эталонных оценок абстрактности слов, тематической направленности наборов ключевых слов. Многие понятия являются субъективными, поэтому результаты можно проверить только на основе экспертной оценки.

Использованные методы реализованы без участия семантических сетей и баз знаний, так как целью работы, результаты которой представлены в статье, было проведение анализа именно графовыми и статистическими методами. По этой причине перспективным представляется создание гибридных методик, которые могут использовать методы и средства, описанные в данной работе, вместе с алгоритмами, основанными на базах знаний. Такой подход позволит проводить сравнительный анализ с результатами работы алгоритмов других типов.

Авторы выражают благодарность д-ру физ.-мат. наук, проф. В. А. Васенину за внимание к работе и высказанные в ходе подготовки статьи замечания.

1. Benz D., Korner C., Hotho A., Stumme G., Strohmaier M. One tag to bind them all: Measuring term abstractness in social metadata // In Working Notes of the LWA 2011. Learning, Knowledge, Adaptation, 2011.
2. Allen R., Wu Y. Generality of texts// Digital Libraries: People, Knowledge, and Technology. Vol. 2555 of Lecture Notes in Computer Science / eds by E. Lim, S. Foo, C. Khoo et al. Berlin: Springer, 2010. P. 111–116.
3. Blei D. M. Probabilistic topic models//Commun. ACM. 2012. Vol. 55, N. 4. P. 77–84.
4. Steyvers M., Griffiths T. Probabilistic Topic Models. Lawrence Erlbaum Associates, 2007.
5. Hotho A., Jäschke R., Schmitz C., Stumme G. Information retrieval in folksonomies: Search and ranking // In Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications, ESWC'06. Berlin: Springer-Verlag, 2006. P. 411–426.
6. Wu Ch., Zhou B. Tags are related: Measurement of semantic relatedness based on folksonomy network // Computing and Informatics. 2011. Vol. 30, N. 1. P. 165–185.
7. Borgatti S. P. Centrality and network flow // Social Networks. 2005. N. 27. P. 55–71.
8. Page L., Brin S., Motwani R., Winograd T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

Приложение А

Самые абстрактные по смыслу слова для каждой меры центральности

Для каждого алгоритма выписаны 50 самых абстрактных ключевых слов. Полужирным шрифтом выделены слова, которые, по мнению авторов, не должны попадать в список самых абстрактных в рамках исследуемого корпуса слов, т. е. ошибочно определенные слова.

- **Betweenness Centrality:** моделирование, модель, структура, оптимизация, математическая модель, математическое моделирование, управление, **мониторинг**, образование, прогнозирование, эксперимент, **прочность**, методы, методика, **самоорганизация**, история, **адаптация**, **здоровье**, **синтез**, анализ, **эффективность**, свойства, диагностика, **инновации**, **оценка**, технология, **устойчивость**, безопасность, личность, **надежность**, компьютерное моделирование, **взаимодействие**, динамика, качество, термодинамика, **плазма**, **наночастицы**, развитие, исследование, культура, **лазер**, теория, интеграция, модернизация, **деформация**, **метод конечных элементов**, **конкурентоспособность**, численное моделирование, **студенты**, алгоритм.

- **Closeness Centrality:** модель, моделирование, структура, **оптимизация**, управление, прогнозирование, методика, эксперимент, анализ, математическая модель, методы, математическое моделирование, **мониторинг**, **эффективность**, **надежность**, качество, технологии, **прочность**, расчет, **оценка**, планирование, **инновационная культура**, исследование, инновации, синтез, **устойчивость**, **взаимодействие**, образование, проектирование, безопасность, **обучение**, динамика, свойства, **деформация**, информационная система, **самоорганизация**, **инновационная деятельность**, вероятность, **профессионализм**, эксплуатация, **здоровье**, интеграция, инновационное развитие, кинетика, **температура**, **вуз**, **адаптация**, **работоспособность**, история, алгоритм.

- **Degree Centrality:** моделирование, математическая модель, математическое моделирование, **оптимизация**, модель, образование, управление, структура, мониторинг,

личность, **прочность**, инновации, свойства, прогнозирование, **эффективность**, **синтез**, методика, культура, **метод конечных элементов**, безопасность, **оценка**, компьютерное моделирование, **наночастицы**, **развитие**, **адаптация**, эксперимент, **студенты**, **здоровье**, качество, история, **анизотропия**, **надежность**, технология, **компетентный подход**, **инновационная деятельность**, численное моделирование, диагностика, модернизация, разрушение, **конкурентоспособность**, творчество, интеграция, **высшая школа**, **компетенции**, **самоорганизация**, **устойчивость**, динамика, **вуз**, **остаточные напряжения**, кинетика.

- **Eigenvector Centrality:** образование, управление, модель, инновации, моделирование, **эффективность**, **инновационная деятельность**, наука, **личность**, методика, **оптимизация**, модернизация, технологии, прогнозирование, мониторинг, компетенции, государство, конкурентоспособность, структура, развитие, интеграция, математическая модель, качество, **оценка**, анализ, история, высшая школа, культура, взаимодействие, студенты, **надежность**, инновационное развитие, методы, **власть**, **бизнес**, **вуз**, стратегия, **компетенция**, эксперимент, **инновационная культура**, обучение, планирование, **бакалавриат**, общество, **компетентный подход**, **здоровье**, **инновационный потенциал**, математическое моделирование, концепция, проект.

- **PageRank Centrality:** моделирование, математическая модель, математическое моделирование, **оптимизация**, модель, образование, **мониторинг**, структура, управление, **метод конечных элементов**, прогнозирование, **прочность**, **наночастицы**, компьютерное моделирование, **личность**, **эффективность**, инновации, развитие, диагностика, численное моделирование, методика, безопасность, **компетентный подход**, культура, **синтез**, **адаптация**, свойства, **здоровье**, **оценка**, **устойчивость**, технология, **надежность**, **разрушение**, **наноструктуры**, **студенты**, интеграция, история, **роман**, динамика, **ани-**

зотропия, профессиональное образование, кинетика, алгоритм, плазма, вуз, конкурентоспособность, каче-

ство, качество образования, остаточные напряжения, дистанционное обучение.

Приложение Б

Найденные в коллекции документов тематические теги

Полужирным шрифтом выделены те теги, которые определены верно:

эпр, медь, алтай, **аудит**, музей, поиск, **право**, доходы, охрана, смазка, стресс, тьютор, услуги, **физика**, катализ, матрица, порошок, контекст, покрытия, преграда, адсорбция, **биометрия**, коррекция, облучение, **семантика**, **кинематика**, **статистика**, предприятие, детали машин, станки с чпу, тестирование, фитопланктон, гидродинамика, дальний восток, самореализация, **конструирование**, диоксид циркония, жидкие кристаллы, пограничный слой, **факторный анализ**, **массовая культура**, преподаватель вуза, имитационная модель, управление знаниями, **нелинейные колебания**, **регрессионный анализ**, **электронное обучение**, ресурсное обеспечение, электроэнцефалограмма, **оптимальное управление**, **физическое моделирование**, образовательная программа, образовательные технологии, поддержка принятия решений, высокоскоростное соударение, **педагогическая деятельность**, международное сотрудничество, научно-образовательный центр, профессиональные компетенции, система менеджмента качества, экспериментальные исследования, **нелинейные динамические системы**, **финансово-хозяйственная деятельность**, федеральный государственный образовательный стандарт, nanoparticles.

Некоторые теги не определяют название дисциплины или направления, но по ним также можно понять тема-

тику документа. Поэтому считается разумным отнести к правильно определенным тематическим тегам следующие:

охрана, покрытия, коррекция, облучение, детали машин, дальний восток, самореализация, управление знаниями, ресурсное обеспечение, образовательная программа, образовательные технологии, профессиональные компетенции, система менеджмента качества.

Далее представлены результаты работы программной реализации алгоритма на данных из Веб:

trade, **testing**, **principal component analysis**, mechanical properties, microstructure, heterogeneity, identification, globalization, **semantic web**, turkey, australia, sensors, information, oxidative stress, wireless sensor networks, tracking, **privacy**, **sustainable development**, **architecture**, feature extraction, obesity, apoptosis, conservation, **pattern recognition**, **risk assessment**, **kinetics**, poverty, india, depression, **cryptography**, climate, diagnosis, virtual reality, parameter estimation, gene expression, collaboration, **policy**, chaos, detection, finite element method, breast cancer, copper, **optimal control**, algorithms, mems, memory, decomposition, concrete, xml, usa, corrosion, taxonomy, **dynamic programming**, planning, volatility, aggregation, **spectroscopy**, russia, **dynamics**, density, mobility, dna, **cfid**, **sensitivity analysis**.

Аналогично случаю с чистыми данными можно дополнить список следующими словами:

mechanical properties, microstructure, wireless sensor networks, virtual reality.

S. A. Afonin, Associate Professor, Researcher, e-mail: serg@msu.ru, Moscow State University, Institute of Mechanics,
K. V. Lunev, Postgraduate Student, e-mail: kirilllunev@gmail.com, Institute of Mechanics Lomonosov Moscow State University

Topic Analysis in Collection of Keyword Tuples

The paper presents algorithms for topic identification in collection of keywords tuples. The analysis of the keywords, namely the degree of their abstractness, is also carried out. The methods used in the paper are based on graph algorithms. The document is presented by the tuple of its keywords. The graph of keywords is built according to the given set of documents. The nodes of the graph fit the keywords, and the edges show the fact of occurrence of words in one tuple. With the help of the graph thematic relationship between documents and thematic characteristics of a document are identified.

Keywords: topic modelling, natural language processing, graph algorithms

References

1. **Benz D., Korner C., Hotho A., Stumme G., Strohmaier M.** One tag to bind them all: Measuring term abstractness in social metadata. In *Working Notes of the LWA 2011*. Learning, Knowledge, Adaptation, 2011.
2. **Allen R., Wu Y.** Generality of texts. *Digital Libraries: People, Knowledge, and Technology*. Vol. 2555 of Lecture Notes in Computer Science / eds by E. Lim, S. Foo, C. Khoo etc. Berlin: Springer, 2010. P. 111–116.
3. **Blei D. M.** Probabilistic topic models. *Commun. ACM*. 2012. Vol. 55, N. 4. P. 77–84.

4. **Steyvers M., Griffiths T.** *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007.

5. **Hotho A., Jaschke R., Schmitz C., Stumme G.** Information retrieval in folksonomies: Search and ranking. In *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications, ESWC'06*. Berlin: Springer-Verlag, 2006. P. 411–426.

6. **Wu Ch., Zhou B.** Tags are related: Measurement of semantic relatedness based on folksonomy network. *Computing and Informatics*. 2011. Vol. 30, N. 1. P. 165–185.

7. **Borgatti S. P.** Centrality and network flow. *Social Networks*. 2005. N. 27. P. 55–71.

8. **Page L., Brin S., Motwani R., Winograd T.** The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

Приближенное решение плохо обусловленных систем линейных алгебраических уравнений с неточными коэффициентами

Предложены оценки результатов приближенных решений плохо обусловленных систем линейных алгебраических уравнений (СЛАУ) некорректно поставленных задач, возникающих при формировании систем с неточными элементами матриц и правых частей.

Показано, что решения систем, сформированных по "знаковой" методике, аналогичны алгебраическому подходу оценки угловых векторов (бруса) объединенного множества решений интервальных СЛАУ (ИСЛАУ) с интервальными матрицами. Рассмотрена связь между возникновением значительных погрешностей решения СЛАУ и применением "знаковой" методики, инициирующей плохо обусловленные матрицы, приводящие к некорректной задаче. В целях получения решений СЛАУ, устойчивых к малым изменениям входных данных, используются операторы регуляризации, численный алгоритм которых развит на основе концепции А. Н. Тихонова по уточнению определения корректности задачи и построению приближенного решения.

Ключевые слова: линейные алгебраические уравнения, "знаковая" методика, интервальная система, оператор регуляризации, некорректная задача, приближенное решение

Введение

Многие задачи, встречающиеся в прикладной математике и физике, технике и управлении, описываются плохо обусловленными системами линейных алгебраических уравнений (СЛАУ). В работе [1] было показано, что подход к определению решений СЛАУ с максимальными отклонениями векторов по "знаковой" методике [2] вполне вписывается в общую схему интервальных оценок угловых векторов, формирующих брус объединенного множества решений интервальных СЛАУ (ИСЛАУ) [3], если коэффициенты СЛАУ представить интервалами с границами, определяющими пределы изменения реальных физических измерений. Применение "знаковой" методики к системам даже сравнительно небольшой размерности ($n \leq 10^2$), с заполненными матрицами, с неточными коэффициентами, может привести к очень плохо обусловленным матрицам и даже разрыву обратных операторов, что инициирует значительную погрешность решения. Реальность возникновения разрывных обратных операторов при вполне непрерывных операторах A приводит к постановке некорректной задачи, точное решение которой не представляется возможным. В силу существования многих задач, не подпадающих под определение корректности по Адамару, А. Н. Тихоновым были сделаны уточнения определения корректности задачи [4, 5]. Концепция

Тихонова относительно решения некорректных задач позволяет построить приближенное решение, устойчивое к малым изменениям входных данных. Вместо решения некорректной задачи, выраженной системой $Ax = b$ (x и b — элементы соответствующих функциональных пространств), обратный оператор которой A^{-1} не непрерывен в метрике бесконечномерного пространства, проводится поиск решения другой системы, принадлежащей конечномерному пространству. На принципах вариационного исчисления, в основе которых лежит минимизация функционала невязок отклонений варьированного решения от точного, строится так называемый регуляризирующий оператор, позволяющий получить приближенное решение системы с неопределенными параметрами. В настоящее время, наряду с методом решения некорректных задач, предложенным А. Н. Тихоновым, дальнейшее развитие получили методы и подходы, изложенные в монографиях М. М. Лаврентьева, В. Я. Арсенина, В. А. Морозова, В. К. Иванова, В. В. Васина и В. П. Тананы, О. А. Лисковца, Г. М. Вайникко, А. Ю. Веретенникова и других авторов. Так, М. М. Лаврентьевым [6] был предложен метод регуляризации сдвигом, а В. К. Иванов [7] обобщил понятие решения с помощью введенного им квази-решения как элемента x , реализующего минимум невязки $\min \{\|Ax - b_\delta\| : x \in Q\} = \|Ax_\delta - b_\delta\|$ на компактном множестве Q (δ — уровень погрешности), что при-

водит к устойчивому обобщенному решению. Такие оценки состоятельны и для операторов A , заданных с погрешностью.

Соотнесение оценки решений СЛАУ по "знаковой" методике и внешней оценки множества решений интервальной задачи

В интервальном виде система линейных алгебраических уравнений, имеющая двухсторонние пределы изменения коэффициентов и вектора воздействия, может быть записана формулой

$$Ax = b \tag{1}$$

с интервальной квадратной матрицей $A \in \mathbb{R}^{n \times n}$ с элементами $a_{ij} \in \mathbb{R}$, $i, j = \overline{1, n}$ где n — число строк и столбцов; $b \in \mathbb{R}^n$ ($\mathbb{R}^{n \times n}$, \mathbb{R}^n и \mathbb{R} — множества интервальных матриц размера $n \times n$, n -мерных векторов и интервалов соответственно); x — искомый (неизвестный) вектор с компонентами x_j размера n . Матрица, правая часть, ее элементы и компоненты представляются через свои интервальные границы:

$$\begin{aligned} A &= [\underline{A}, \overline{A}]; \quad b = [\underline{b}, \overline{b}] \quad \text{и} \quad a_{ij} = [\underline{a}_{ij}, \overline{a}_{ij}], \\ b_i &= [\underline{b}_i, \overline{b}_i], \quad \forall i, j = \overline{1, n}, \quad a_{ij} = [a_{ij} - \varepsilon_{ij} a_{ij}, a_{ij} + \varepsilon_{ij} a_{ij}], \\ b_i &= [b_i - \delta_i b_i, b_i + \delta_i b_i], \quad \forall i = \overline{1, n}, \quad \forall j = \overline{1, n}, \\ \varepsilon_{ij} &\leq |\varepsilon_0|, \quad \delta_i \leq |\delta_0|, \end{aligned} \tag{2}$$

где a_{ij} и b_i — средние значения интервалов; ε_{ij} и δ_i — относительные погрешности элементов матрицы и компонент правых частей, ограниченные сверху заданными значениями ε_0 и δ_0 .

С позиций интервальной математики система (1) является совокупностью всех точечных систем размера $n \times n$

$$Ax = b, \tag{3}$$

где точечные матрицы $A \in \mathbb{R}^{n \times n}$ и векторы $b \in \mathbb{R}^n$, относящиеся к множеству вещественных матриц и векторов, принадлежат интервальной матрице и вектору правых частей: $A \in \mathbf{A}$ и $b \in \mathbf{b}$.

Постановка задачи об определении "внешних" покоординатных оценок множества решений записывается в следующем виде:

$$\Sigma = \{x \in \mathbb{R}^n \mid (\exists A \in \mathbf{A})(\exists b \in \mathbf{b})(Ax = b)\}. \tag{4}$$

Множество (4) образовано всеми решениями точечных систем $Ax = b$, где $A \in \mathbf{A}$ и $b \in \mathbf{b}$, что соответствует нижней и верхней оценкам $\min \{x_k \mid x \in \Sigma\}$ и $\max \{x_k \mid x \in \Sigma\}$, $k = \overline{1, n}$ бруса, определенного декартовым произведением максимальных интервалов решения x . В общем случае больших размерностей системы требуется найти некоторый гипербрус (со сторонами, параллельными координатным осям), содержащий множество решений (4).

Аналогию в постановке интервальной и алгебраической задач можно проследить, во-первых, в результате представления СЛАУ в виде интервальной системы, когда каждый из элементов матрицы и компонент правой части выражается интервалом с границами относительно своих средних значений (2). Во-вторых, по результатам оценки решений для максимальных значений x , которые возможны при заданных параметрах алгебраической системы. "Знаковый" подход позволяет сконструировать новые границы интервальной матрицы A и вектора b : $A = [A^-, A^+]$, $b = [b^-, b^+]$, $A^{-1} = [(A^-)^{-1}, (A^+)^{-1}]$ с

элементами a_{ij}^- и a_{ij}^+ и граничными векторами $b^- = (b_1^-, b_2^-, \dots, b_n^-)$ и $b^+ = (b_1^+, b_2^+, \dots, b_n^+)$. Матрицы A^+ и A^- и векторы b^+ и b^- отличны от граничных матриц \overline{A} и \underline{A} и векторов \overline{b} и \underline{b} , так как они формируются отбором элементов в соответствии со знаками относительных погрешностей ε_0 и δ_0 . Отбор элементов матрицы и правых частей по знаку относительной погрешности проводится в соответствии с принципом наименьших и наибольших отклонений определителей матриц $\det(A^+)$ или $\det(A^-)$ от их среднего значения. Наибольшие изменения значений определителей $\det(A^+)$ и $\det(A^-)$ в положительном и отрицательном направлениях вызывают максимальные изменения компонент вектора решения. На основании проведенного анализа формируются две матричные системы

$$\begin{cases} A^- x = b_j^+ \\ A^+ x = b_j^- \end{cases}, j = \overline{1, n} \quad \text{и} \quad \begin{cases} A^- x = b_j^- \\ A^+ x = b_j^+ \end{cases}, j = \overline{1, n}.$$

На основании этих систем можно получить три основные матричные системы, подозрительные на оптимальные граничные значения векторов x с $2n$ компонентами каждый:

$$\begin{cases} A_{a_j}^- x = b_j^- \\ A_{a_j}^+ x = b_j^+ \end{cases}, j = \overline{1, n}, \quad \begin{cases} A_{a_j}^+ x = b_j^+ \\ A_{a_j}^- x = b_j^- \end{cases}, j = \overline{1, n}, \tag{5}$$

$$\begin{cases} A_{a_j}^- x = b_j \\ A_{a_j}^+ x = b_j \end{cases}, j = \overline{1, n}.$$

Расчеты векторов x , аналогичные расчетам через обратные матрицы системы (5), могут быть получены по формулам Крамера:

$$\begin{aligned} x_j^- &= \frac{\det(A_{b_j}^-)}{\det(A_a^-)}, \quad x_j^- = \frac{\det(A_{b_j}^+)}{\det(A_a^+)}, \quad x_j^+ = \frac{\det(A_{b_j}^+)}{\det(A_a^-)}, \\ x_j^+ &= \frac{\det(A_{b_j}^-)}{\det(A_a^-)}, \quad x_j^- = \frac{\det(A_b^-)}{\det(A_a^+)}, \quad x_j^+ = \frac{\det(A_b^+)}{\det(A_a^-)}. \end{aligned} \tag{6}$$

Индексы элементов, входящих в детерминанты формул (6), определяют замену столбцов в матрицах

A^+ и A^- на соответствующие столбцы из противоположных матриц A^- и A^+ . Так, например, матрица $A_{a_j}^-$ получена заменой j -го столбца матрицы A^- аналогичным столбцом из матрицы $A^+ - a_j^+$; матрица $A_{b_j}^-$ получена заменой j -го столбца матрицы A^- вектором правой части b_j^+ .

Искомый вектор $x = [x^-, x^+]$ с граничными векторами

$$x^- = (x_1^-, x_2^-, \dots, x_n^-), \quad x^+ = (x_1^+, x_2^+, \dots, x_n^+) \quad (7)$$

формируется выборкой $2n$ наибольших и наименьших компонент векторов x^-, x^+ из расчетов по формулам (6). Верхние символы $(-)$ и $(+)$ характеризуют максимальные граничные отклонения решения x от средних значений вправо и влево. Граничные векторы x^-, x^+ соответствуют нижним и верхним оценкам $\min\{x_k \mid x \in \Sigma\}$ и $\max\{x_k \mid x \in \Sigma\}$, $k = \overline{1, n}$ бруса множества решений интервальной системы.

Реализация внешнего интервального оценивания множества решений системы размера $n \times n$ составляет $2n \cdot 2^{n^2}$ операций, а по "знаковой" методике сводится к решению $3 \cdot 2n$ алгебраических формул (6) с дальнейшей выборкой максимальных и минимальных компонент.

Дополнительные трудности для сравнения интервальной и "знаковой" оценок максимальных отклонений, сопровождающиеся увеличением числа уравнений системы (5), представляют случаи независимого задания границ интервалов матриц \tilde{A} и \tilde{A} , имеющих разные знаки.

Аналогия между интервальной задачей и точечной задачей нарушается в случае приближения относительных погрешностей ε_0 и δ_0 к 100%.

Сингулярность решения возникает при приближении к нулю любого из определителей, входящих в детерминанты знаменателей формулы (6), и зависит от абсолютной величины ε_0 и δ_0 . Проявление сингулярности матрицы СЛАУ всегда обуславливает некорректность задачи, но обнаруживается и без возникновения сингулярности в системах с плохо обусловленными матрицами.

Основным требованием к корректности задачи является соблюдение условия незначительных отклонений решения при малых возмущениях параметров исходной системы. Для случая рассматриваемых СЛАУ с формализованными матрицами A^+ и A^- и правыми частями b^+ и b^- некорректность задачи проявляется в том, что при малых по норме отклонениях операторов и правых частей системы (5) (ε и δ много меньше 1) $\|A^+ - A\| \leq \varepsilon$, $\|A^- - A\| \leq \varepsilon$, $\|b^+ - b\| \leq \delta$, $\|b^- - b\| \leq \delta$, решения x получают значительные отклонения Δ , нарушая тем самым третье условие кор-

ректности задачи по Адамару: $\|x^+ - x\| \geq \Delta$, $\|x^- - x\| \geq \Delta$. Следует отметить, что при конструировании матриц A^+ и A^- и правых частей b^+ и b^- значительные выбросы погрешностей решения наблюдаются и для отдельных компонент векторов x^+ и x^- .

Получение приближенных решений некорректных задач выбором оператора регуляризации

При изучении систем (5) проводится исследование матричных уравнений первого рода $Ax = b$ с заданным оператором A отображения из X в Y , определяющих решения в линейных нормированных пространствах $X(x \in X)$ и $Y(b \in Y)$ (в общем случае X и Y — произвольные топологические пространства). В случае некорректно поставленной задачи нельзя получить точное решение с приближенными правой частью \tilde{b} и матрицей \tilde{A} . Понятие приближенного решения \tilde{x} вводится с помощью так называемого регуляризующего оператора R_α , зависящего от параметра α , определенного для всех $\alpha > 0$ и всех \tilde{b} и \tilde{A} , входящих в класс допустимых исходных данных $X(A)$, $Y(b)$. Решение $\tilde{x} = R_\alpha(\tilde{A}, \tilde{b})$ уклоняется от x меньше чем на заданную точность ε : $\rho(x, \tilde{x}) \leq \varepsilon$ (ρ — мера уклонения приближенного решения от точного). Построенный оператор регуляризации R_α позволяет получить приближенное решение $\tilde{x}_\alpha = x_0 + (\alpha E + \tilde{A}^T \tilde{A})^{-1} \tilde{A}^T (\tilde{b} - \tilde{A}x_0)$ (E — единичная диагональная матрица; x_0 — начальное приближение вектора \tilde{x}_α).

На основании методов градиентного типа [8] может быть получена устойчивая аппроксимация решения регуляризованного уравнения, которое для системы (5) оказывается более приемлемым в плане получения приближенного решения

$$x^{k+1} = x^k - \frac{\left(\text{norm}(A^T A x^k - A^T b)\right)^2}{\left(\text{norm}(A(A^T A x^k - A^T b))\right)^2} \times A^T (A x^k - b). \quad (8)$$

При подстановке матриц A^+ и A^- и векторов b^+ и b^- в уравнение (8) получается система из $2n$ уравнений за счет двух наборов векторов правых частей $b = [b^-, b^+]$, $b^- = \{b_j^-\}$, $b^+ = \{b_j^+\}$, $j = \overline{1, n}$, позволяющая вычислить приближенные решения x_α^+ и x_α^- , корректирующие x^+ и x^- , полученные по "знаковой" методике. В качестве начального приближения принимаем вектор решения средней системы $x_0 = (x_1, x_2, \dots, x_n)$:

$$\begin{cases} x_\alpha^+ = x_0 - \left(\text{norm}\left(\left(A^-\right)^T A^- x_0^+ - \left(A^-\right)^T b^+\right)\right)^2 / \left(\text{norm}\left(A^-\left(\left(A^-\right)^T A^- x_0^{+(k)} - \left(A^-\right)^T b^+\right)\right)\right)^2 \left(A^-\right)^T \left(A^- x_0^+ - b^+\right) \\ x_\alpha^- = x_0 - \left(\text{norm}\left(\left(A^+\right)^T A^+ x_0^- - \left(A^+\right)^T b^-\right)\right)^2 / \left(\text{norm}\left(A^+\left(\left(A^+\right)^T A^+ x_0^- - \left(A^+\right)^T b^-\right)\right)\right)^2 \left(A^+\right)^T \left(A^+ x_0^- - b^-\right) \end{cases} \quad (9)$$

На основании формул (9), как итерационных операторов регуляризации, вычисляются $2n$ векторов $x_\alpha = [x_\alpha^-, x_\alpha^+]$ с n компонентами для каждого вектора x_α^- и x_α^+ , отвечающие двусторонним границам регуляризации.

Примеры приближенных решений путем использования оператора регуляризации

В данном разделе приведены решения некорректных задач, представленных алгебраическими системами с плохо обусловленными матрицами, вызванными относительной погрешностью интервальных коэффициентов. На примере СЛАУ десятого порядка проводится сравнение абсолютных и относительных отклонений от средних значений решений, полученных из уравнений регуляризации (9) и граничных векторов (7).

Рассмотрим систему десятого порядка со случайными элементами и проследим на ее примере возникновение погрешности решения в зависимости от интервала неопределенности ее коэффициентов, выраженного симметричными интервалами относительно средних значений параметров системы. Для удобства записи система изображена для средних значений матрицы и правой части, интервальные отклонения коэффициентов системы представляются через относительные погрешности ε_0 и δ_0 (2). Через вариации погрешностей ε_0 и δ_0 прослеживается характер изменения двухсторонних граничных решений системы, полученных по "знаковой" методике (7), и приближенных решений, вычисленных с помощью операторов регуляризации (9). Проводится оценка их абсолютных и относительных отклонений по отношению к решениям средней системы. В качестве примеров приведены три варианта расчетов.

Вариант 1. $\varepsilon_0 = 0,001$, $\delta_0 = 0,001$.

$$A = \begin{pmatrix} 0,5144 & 0,5341 & 0,8507 & 0,6126 & 0,7386 & 0,7690 & 0,5523 & 0,1465 & 0,1239 & 0,9479 \\ -0,8843 & 0,0900 & 0,5606 & 0,9900 & 0,5860 & 0,5814 & 0,6299 & 0,1891 & 0,4904 & 0,0821 \\ 0,5880 & 0,1117 & 0,9296 & 0,5277 & 0,2467 & 0,9283 & 0,0320 & 0,0427 & 0,8530 & 0,1057 \\ 0,1548 & -0,1363 & 0,6967 & 0,4795 & 0,6664 & 0,5801 & 0,6147 & 0,6352 & 0,8739 & 0,1420 \\ 0,1999 & 0,6787 & 0,5828 & 0,8013 & 0,0835 & 0,0170 & 0,3624 & 0,2819 & 0,2703 & 0,1665 \\ 0,4070 & 0,4952 & 0,8154 & 0,2278 & 0,6260 & 0,1209 & 0,0495 & 0,5386 & 0,2085 & 0,6210 \\ 0,7487 & 0,1897 & 0,8790 & -0,4981 & 0,6609 & 0,8627 & 0,4896 & 0,6952 & -0,5650 & 0,5737 \\ 0,8256 & 0,4950 & 0,9889 & 0,9009 & 0,7298 & 0,4843 & 0,1925 & 0,4991 & 0,6403 & 0,0521 \\ 0,7900 & 0,1476 & 0,0005 & 0,5747 & 0,8908 & -0,8449 & 0,1231 & 0,5358 & 0,4170 & 0,9312 \\ 0,3185 & 0,0550 & 0,8654 & 0,8452 & 0,9823 & 0,2094 & 0,2055 & 0,4452 & 0,2060 & 0,7287 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_0 \end{pmatrix} = \begin{pmatrix} 0,7904 \\ 0,9493 \\ 0,3276 \\ 0,6713 \\ 0,4386 \\ 0,8335 \\ 0,7689 \\ 0,1673 \\ 0,8620 \\ 0,9899 \end{pmatrix}.$$

На рис. 1 представлены результаты расчетов (листинг), проведенных по программе, разработанной для данной задачи и формализованной в системе MATLAB [9]. Все исходные данные приняты безразмерными величинами.

В первом столбце приведены результаты вычислений (в безразмерных единицах) средней системы (10), во втором и четвертом столбцах — граничные отклонения, полученные по "знаковой" методике вправо и влево соответственно; в третьем и пятом столбцах — их приближенные значения при использовании оператора регуляризации. В шестом и седьмом столбцах даны относительные погрешности граничных отклонений, в восьмом и девятом столбцах — приближенные относительные отклонения при использовании оператора регуляризации (в процентах).

Вычисленные определители средней и граничных матриц: $\det(A_c) = 0,0204$; $\det(A^+) = 0,0244$; $\det(A^-) = 0,0164$. Начальные отклонения по норме для матриц A^+ и A^- и векторов b^+ , b^- от их средних значений равны следующим значениям: $\text{norm}(A^+ - A_c) = 0,0050$; $\text{norm}(A^- - A_c) = 0,0050$; $\text{norm}(b^+ - b_c) = 0,0047$; $\text{norm}(b^- - b_c) = 0,0046$. Отклонения по норме выборки граничных значений x от среднего: $\text{norm}(x^+ - x_c) = 2,7533$ и $\text{norm}(x^- - x_c) = 2,5529$. Приведенные числовые значения показывают, что оценки отклонений граничных решений по норме превосходят в тысячу раз оценки исходных данных, что является явным нарушением корректности задачи. Применение оператора регуляризации для системы (10) обеспечивает получение решения $x_{aj} = [x_{aj}^-, x_{aj}^+]$ для каждого столбца n -мерных векторов b_j^- и b_j^+ , где символ j относится к отдельной компоненте вектора и ее знаковому наполнению при подстановке в j -е столбцы матриц A^- и A^+ по формулам (6). Отклонения по нормам для приближенного решения в зависимости от правой части системы (10) занесены в таблицу (все расчетные данные, как и исходные, являются безразмерными величинами). Во всех случаях вычислений норм, как и в данном случае, принята евклидова норма.

Сравнение оценок отклонений по нормам рас-

считанных приближенных векторов показывает, что они практически не меняются от столбца к столбцу. Как следствие, с вычисленными нормами разности приближенных и средних значений $\text{norm}(x_{aj}^+ - x_c)$ и $\text{norm}(x_{aj}^- - x_c)$ приближенное решение x_α может быть

Решение средн.	MAX откл. вправо	Регуляриз. по Тихонову вправо	MIN откл. влево	Регуляриз. по Тихонову влево	Погрешность откл. решения влево (%)	Погрешность откл. решения вправо (%)	Попр. погрешн. по Тихонову (%) влево	Попр. погрешн. по Тихонову (%) вправо
-0.85462	-0.794247	-0.854201	-0.952856	-0.855	7.06433	11.4947	0.0444587	0.0490581
0.236063	0.347642	0.235943	0.159562	0.236172	47.2669	32.4067	0.0465586	0.0504565
-5.05767	-4.0939	-5.05639	-6.48585	-5.05889	19.0556	28.2377	0.024098	0.0253059
3.31367	4.16545	3.31599	2.73947	3.31128	25.705	17.3281	0.0722385	0.0698954
-2.961	-2.45998	-2.96013	-3.70525	-2.96181	16.9208	25.135	0.027458	0.0292856
4.81142	6.0784	4.81157	3.95702	4.81119	26.3329	17.7577	0.00472342	0.00317088
-4.28195	-3.53426	-4.28444	-5.3912	-4.27944	17.4614	25.9052	0.0585852	0.0580605
7.28378	9.03413	7.2822	6.10351	7.2853	24.0309	16.204	0.0208483	0.0217278
-0.602662	-0.471876	-0.600084	-0.796023	-0.605192	21.7014	32.0846	0.419946	0.427732
3.41394	4.05753	3.41301	2.9805	3.41481	18.8516	12.6962	0.0254529	0.0273838

Рис. 1. Результаты расчетов средних и граничных решений, абсолютных и относительных отклонений влево и вправо граничных и приближенных компонент векторов решения от средних значений; $\varepsilon_0 = 0,001$, $\delta_0 = 0,001$

получено при любой подстановке векторов b_j^+ и b_j^- в матрицы A^- и A^+ .

В качестве параметра регуляризации принята величина α , значение которой близко к 10^{-5} . В качестве начальной подстановки принимается x_0 . Остановливаясь уже на первой или второй итерации можно получить удовлетворительные оценки приближенных векторов по нормам, компоненты которых незначительно изменяются влево и вправо при дальнейших итерациях.

На рис. 2 (см. третью сторону обложки) показана зависимость безразмерных компонент вектора решения от номера компоненты N : a — среднего x_c , граничных x^+ и x^- и приближенных $x_r = \{x_\alpha^-, x_\alpha^+\}$; b — правой части b^+ и b^- . На рис. 3 (см. третью сторону обложки) и рис. 4 показаны абсолютные $\Delta x = \{\Delta x_{\min}, \Delta x_{\max}\}$,

$$\Delta x_{\max} = (x^+ - x_c), \quad \Delta x_{\min} = (x^- - x_c), \quad \Delta x_r = \{\Delta x_\alpha^-, \Delta x_\alpha^+\},$$

$$\Delta x_\alpha^- = (x_\alpha^- - x_c), \quad \Delta x_\alpha^+ = (x_\alpha^+ - x_c), \quad \Delta x_\alpha^- = (x_\alpha^- - x_c) \text{ и отно-}$$

$$\text{сительные } \frac{\Delta x}{x_c} = \left\{ \frac{\Delta x_{\min}}{x_c}, \frac{\Delta x_{\max}}{x_c} \right\}, \quad \frac{\Delta x_{\max}}{x_c} = (x^+ - x_c) / x_c,$$

$$\frac{\Delta x_{\min}}{x_c} = (x^- - x_c) / x_c, \quad \frac{\Delta x_r}{x_c} = \left\{ x_\alpha^- / x_c, x_\alpha^+ / x_c \right\}, \quad \frac{\Delta x_\alpha^+}{x_c} =$$

$$= (x_\alpha^+ - x_c) / x_c, \quad \frac{\Delta x_\alpha^-}{x_c} = (x_\alpha^- - x_c) / x_c \text{ отклонения (в про-}$$

центах) граничных и приближенных решений от среднего в зависимости от номера компоненты N . На рис. 3, б, 4, б (и далее в статье на рис. 6, б, 7, б, 9, б 10, б) разными цветами показаны в увеличенном масштабе кривые относительных отклонений двух близких приближенных решений $\frac{x_\alpha^-}{x_c}$ и $\frac{x_\alpha^+}{x_c}$.

Вариант 2. $\varepsilon_0 = 0,0051384$; $\delta_0 = 0,0051384$.

Возрастание относительных погрешностей ε_0 и δ_0 приближает один из детерминантов системы (6) к нулю и при дальнейшем их увеличении — к переходу определителя через ноль и смене его знака, провоцируя тем самым значительные отклонения граничных решений и значения их относительных и абсолютных отклонений. На рис. 5 даны результаты вычислений (листинг), включая два последних столбца с относительными отклонениями приближенных решений при использовании операторов регуляризации.

Вычисленные определители равны следующим значениям: $\det(A_c) = 0,0204$; $\det(A^+) = 0,0408$; $\det(A^-) = 1,4599 \cdot 10^{-7}$. Отклонения по норме для матриц A^+ и A^- и векторов b^+ , b^- от их средних значений: $\text{norm}(A^+ - A_c) = 0,0259$; $\text{norm}(A^- - A_c) = 0,0259$; $\text{norm}(b^+ - b_c) = 0,0118$; $\text{norm}(b^- - b_c) = 0,0118$; отклонения оптимальных граничных решений от среднего по норме: $\text{norm}(x^+ - x_c) = 1,3951 \cdot 10^6$; $\text{norm}(x^- - x_c) = 1,1370 \cdot 10^6$. С устремлением одного из определителей системы к нулю, при бесконечно большом воз-

Отклонения по нормам приближенных решений от среднего решения

j	1	2	3	4	5	6	7	8	9	10
$\text{norm}(x_{aj}^+ - x_c)$	0,0047	0,0046	0,0044	0,0049	0,0044	0,0049	0,0044	0,0049	0,0047	0,0049
$\text{norm}(x_{aj}^- - x_c)$	0,0046	0,0046	0,0044	0,0049	0,0044	0,0049	0,0044	0,0049	0,0047	0,0049

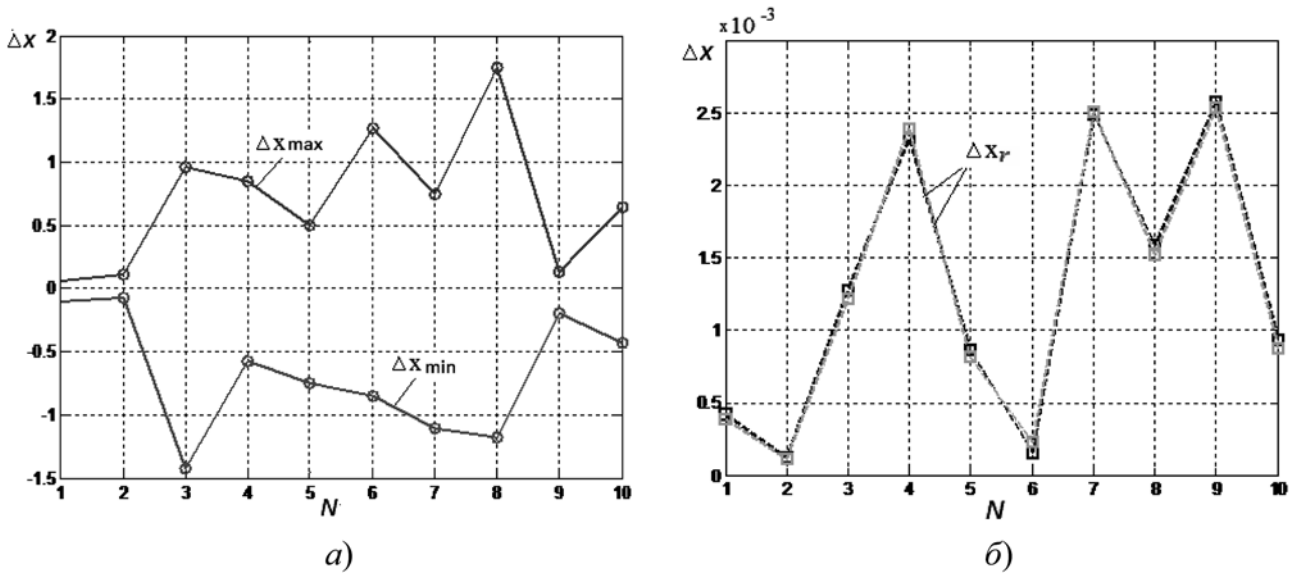


Рис. 4. Зависимость абсолютных отклонений компонент векторов решений от номера компоненты: а — граничные векторы Δx_{\min} , Δx_{\max} ; б — приближенные векторы $\Delta x_r \cdot 10^{-3}$; $\varepsilon_0 = 0,001$, $\delta_0 = 0,001$

Решение средн.	MAX откл. вправо	Регуляриз. по Тихонову вправо	MIN откл. влево	Регуляриз. по Тихонову влево	Погрешность откл. решения влево (%)	Погрешность откл. решения вправо (%)	Попр. погрешн. по Тихонову (%) влево	Попр. погрешн. по Тихонову (%) вправо
-0.85462	-0.667115	-0.852055	-50858.5	-0.856148	21.9401	5.9509e+006	0.178711	0.300205
0.236063	62267.7	0.235344	-0.00691833	0.236539	2.63775e+007	102.931	0.201785	0.304347
-5.05767	-2.09377	-5.05046	-824374	-5.06327	58.6022	1.62994e+007	0.110687	0.142582
3.31367	492734	3.32472	1.55172	3.30057	1.48696e+007	53.1722	0.395442	0.333498
-2.961	-1.42794	-2.95598	-431734	-2.96459	51.775	1.45806e+007	0.121229	0.169501
4.81142	732382	4.81139	2.18771	4.80947	1.52217e+007	54.5309	0.0403693	0.000613812
-4.28195	-1.98812	-4.29449	-641805	-4.26881	53.5698	1.49885e+007	0.306868	0.292898
7.28378	1.01197e+006	7.27495	3.65999	7.29092	1.38933e+007	49.7515	0.0980544	0.121231
-0.602662	-0.198577	-0.588923	-111073	-0.615164	67.05	1.84303e+007	2.07451	2.27961
3.41304	373022	3.40842	2.08651	3.41773	1.09263e+007	38.8828	0.110865	0.161779

Рис. 5. Результаты расчетов средних и граничных решений, абсолютных и относительных отклонений влево и вправо граничных и приближенных компонент векторов решения от средних значений; $\varepsilon_0 = 0,0051384$; $\delta_0 = 0,0051384$

растании граничных значений x^- и x^+ , нормы, как и обратные операторы, стремятся к бесконечности.

Как видно из результатов расчетов, по мере приближения одной из граничных матриц системы (10) к сингулярности граничные оптимальные решения (см. столбцы 2, 4 на рис. 5) и относительные погрешности решений, представленные столбцами 6, 7 на рис. 5, достигают чрезвычайно больших значений. На графиках рис. 6 (см. третью сторону обложки) и рис. 7 видно, что относительные погрешности приближенных решений можно снизить на много порядков, а размер отклонений — в тысячи раз, несмотря на то что определитель одной из матриц системы приближается к нулю.

Вариант 3. $\varepsilon_0 = 0,1$, $\delta_0 = 0,1$.

Воспользуемся операторами регуляризации системы при значительных интервальных отклонениях коэффициентов матрицы и правых частей системы (10), когда неточность интервальных коэффициен-

тов матрицы системы инициирует переход одного из определителей через ноль.

Вычисленные значения норм для отклонений матриц, векторов правой части, граничных оптимальных и приближенных решений:

$$\det(A) = 0,0204; \det(A^+) = 0,4149; \det(A^-) = -0,3702;$$

$$\text{norm}(A^+ - A_c) = 0,5040; \text{norm}(A^- - A_c) = 0,5040;$$

$$\text{norm}(b^+ - b_c) = 0,2305; \text{norm}(b^- - b_c) = 0,2305;$$

$$\text{norm}(x^+ - x_c) = 28,8388; \text{norm}(x^- - x_c) = 38,6403;$$

$$\text{norm}(x_a^+ - x_c) = 0,7017; \text{norm}(x_a^- - x_c) = 0,5932.$$

На рис. 8 показаны покомпонентные результаты расчетов (листинг) оптимальных векторов и их приближений операторами регуляризации.

Из расчетов, показанных на рис. 8 и представленных на рис. 9, 10, видно, что приближенные решения с ростом границ интервалов исходной матрицы и

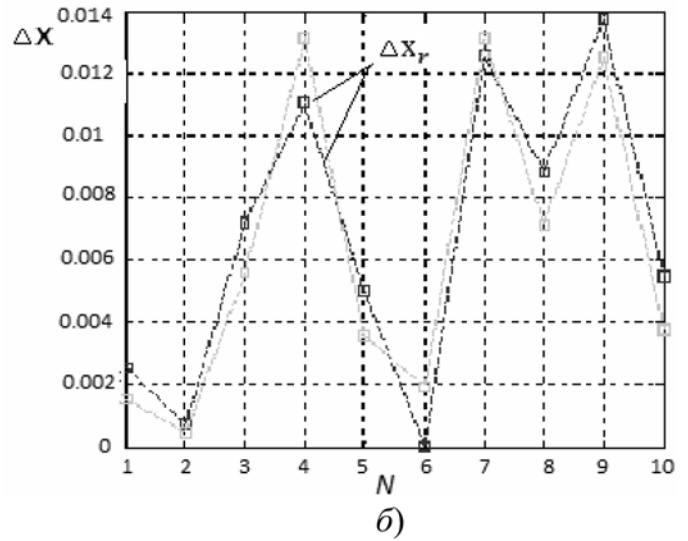
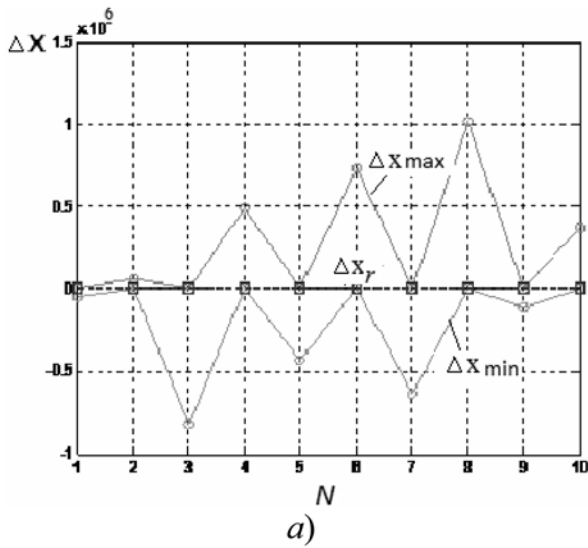


Рис. 7. Зависимость абсолютных отклонений компонент векторов решений от номера компоненты:

а — граничные ΔX_{\min} , ΔX_{\max} и приближенные ΔX_r векторы; б — приближенные векторы ΔX_r ; $\varepsilon_0 = 0,0051384$; $\delta_0 = 0,0051384$

Решение средн.	MAX откл. вправо	Регуляриз. по Тихонову вправо	MIN откл. влево	Регуляриз. по Тихонову влево	Попрешность откл. решения влево (%)	Попрешность откл. решения вправо (%)	Попр. погрешн. по Тихонову (%) влево	Попр. погрешн. по Тихонову (%) вправо
-0.85462	-0.255782	-0.703925	-10.9889	-0.737536	70.0707	1185.82	13.7001	17.633
0.236063	1.01105	0.193791	-5.05438	0.232512	328.297	2241.12	1.50398	17.9069
-5.05767	10.0289	-4.79074	-9.73947	-4.91072	298.29	92.5681	2.90559	5.27785
3.31367	6.05069	3.17981	-4.33681	2.9273	82.5979	230.876	11.6599	4.03961
-2.961	4.97049	-2.73905	-5.22222	-2.81374	267.865	76.3667	4.97333	7.49587
4.81142	8.90184	4.56422	-3.26209	4.58499	85.015	167.799	4.70607	5.13765
-4.28195	9.29977	-4.35179	-7.80846	-4.00651	317.185	82.3574	6.43265	1.63105
7.28378	12.8678	6.97491	-6.48768	7.21702	76.6635	189.07	0.916573	4.24058
-0.602662	1.66909	-0.30259	-2.35791	-0.619269	376.953	291.249	2.75566	49.7911
3.41394	20.5947	3.13458	-28.311	3.29664	503.252	929.274	3.43593	8.18302

Рис. 8. Результаты расчетов средних и граничных решений, абсолютных и относительных отклонений влево и вправо граничных и приближенных компонент векторов решения от средних значений; $\varepsilon_0 = 0,1$, $\delta_0 = 0,1$

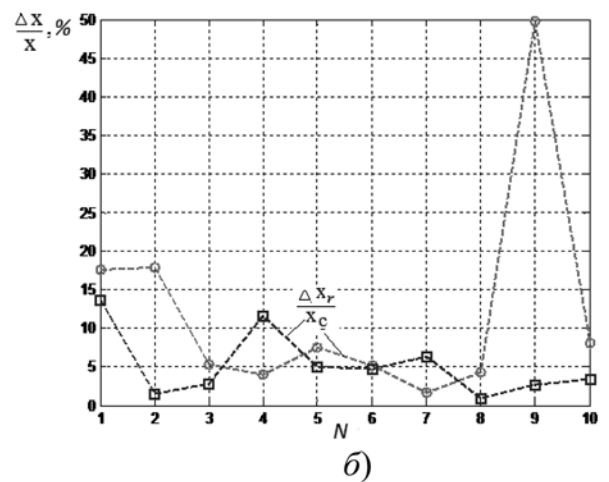
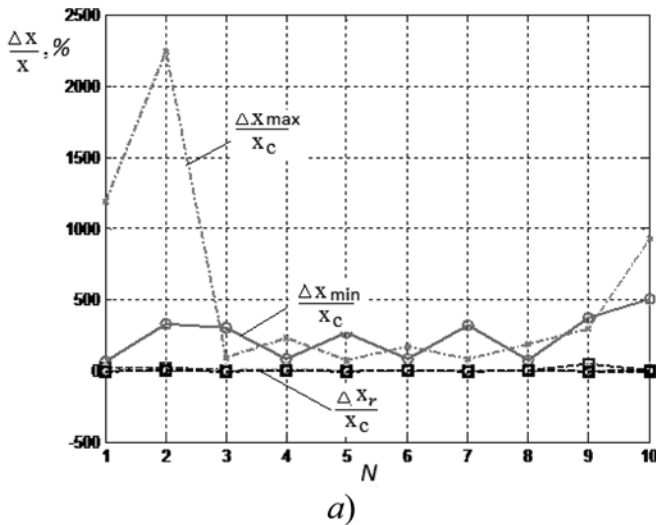


Рис. 9. Зависимость относительных отклонений компонент векторов решений от номера компоненты:

а — граничные $\frac{\Delta X_{\min}}{x_c}$, $\frac{\Delta X_{\max}}{x_c}$ и приближенные $\frac{\Delta X_r}{x_c}$ векторы; б — приближенные векторы $\frac{\Delta X_r}{x_c}$; $\varepsilon_0 = 0,1$, $\delta_0 = 0,1$

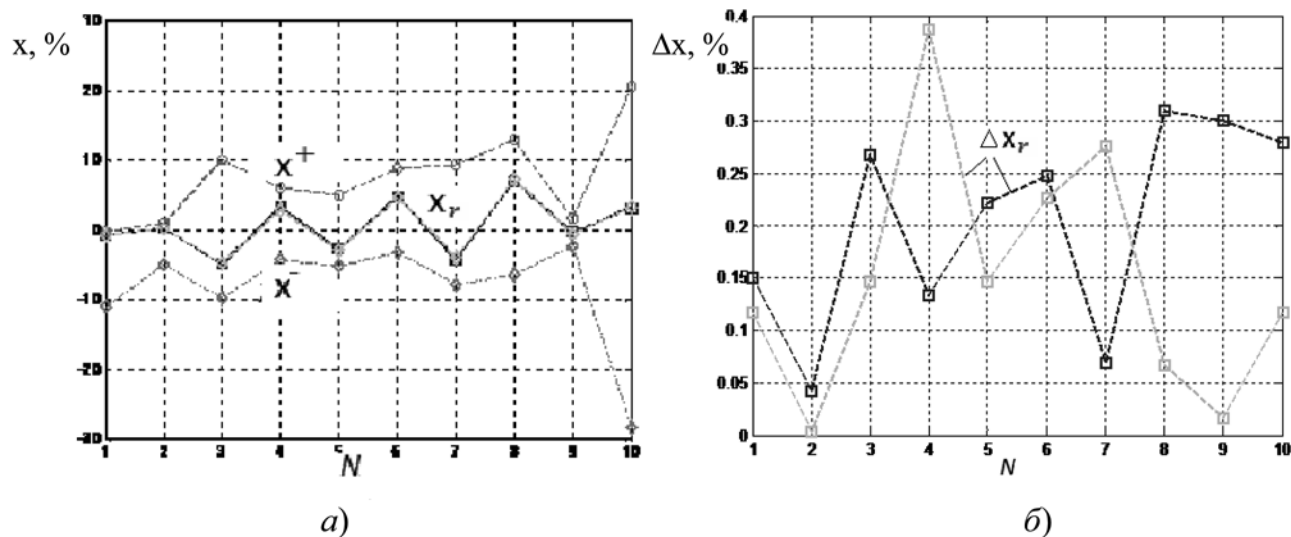


Рис. 10. Зависимость компонент векторов решений от номера компоненты:

a — граничные x^- , x^+ и приближенные x_r векторы; b — абсолютные отклонения приближенных векторов Δx_r ; $\varepsilon_0 = 0,1$, $\delta_0 = 0,1$

правых частей системы (1) имеют тенденцию к большим абсолютным и относительным погрешностям в соответствии с погрешностью исходных данных (матриц и векторов системы). Однако даже при 10%-ных относительных отклонениях, когда измеренное значение попадает в 20%-ный интервал изменения параметров, и переходе одного из определителей через ноль возможно снижение относительной погрешности решения с 2200 до 49 %.

С увеличением порядка системы порог минимума для значения ε , приводящего к смене знака определителей, сильно снижается и приводит к возникновению вырожденных матриц при значительно меньших их абсолютных значениях.

Заключение

Впервые рассмотрена связь между возникновением значительных погрешностей решения СЛАУ с применением "знаковой" методики и плохой обусловленностью матриц, возникающих при этом. Тем самым ставится вопрос о надежности оценок получаемых результатов в плане их точности как решения плохо обусловленных систем. Анализ и проведенные оценки показывают, что значительный рост погрешности решения может иметь объяснение не в силу применения "знаковой" методики при формализации знакопеременных матриц, а в результате некорректности задачи, возникающей в результате нарушения непрерывности обратного оператора системы.

Показано соответствие результатов решения интервальной задачи по определению внешней оценки множества решений и задачи определения максимальных отклонений решения под влиянием знакопеременных относительных погрешностей коэффициентов матрицы и правых частей СЛАУ.

Впервые реализован и объяснен способ преобразования интервальных матриц в знакопеременные, допускающие вычисления максимальных погрешностей решения.

На примере решения системы десятого порядка показано влияние на погрешность решения абсолютной величины исходных погрешностей решения, с возрастанием которых невозможно полагаться на правильность точного решения системы с неточными входными данными.

Наглядность полученных результатов вычислений поддерживается рассмотрением одной и той же системы, приближенные решения которой получены для систем с непрерывным обратным оператором A^{-1} (вариант 1), приближением определителя $\det(A)$ к нулю или при $A^{-1} \rightarrow \infty$ (вариант 2), при переходе определителя системы через ноль и смене знака одного из граничных определителей (вариант 3). Показано, что несмотря на переход через вырожденность определителя граничной матрицы и разрыв обратного оператора, непрерывность решения сохраняется, а алгоритм регуляризации обеспечивает хорошее понижение погрешности решения.

Расчеты проведены на основании программы, включающей совместные алгоритмы вычислений оценок решения системы с неточно задаваемыми параметрами по "знаковой" методике и приближенных решений некорректных задач с помощью операторов регуляризации, основанные на концепции А. Н. Тихонова.

Список литературы

1. Иванова К. Ф. Знаковый подход к оценке решения интервальных линейных систем // Информационные технологии. 2012. № 9. С. 46–52.

2. **Петров Ю. П.** Как обеспечить надежность решения систем уравнений. СПб.: БХВ-СПб, 2009. 172 с.
 3. **Шарый С. П.** Конечномерный интервальный анализ. Новосибирск: XYZ, 2007. 700 с.
 4. **Тихонов А. Н., Арсенин В. Я.** Методы решения некорректных задач. М.: Наука. Главная редакция физико-математической литературы, 1979. 284 с.
 5. **Тихонов А. Н.** О решении некорректно поставленных задач и методе регуляризации // ДАН СССР. 1963. Т. 151, № 3. С. 501—504.

6. **Лаврентьев М. М.** О некоторых некорректных задачах математической физики. Новосибирск: Изд-во СО АН СССР, 1962. 92 с.
 7. **Иванов В. К.** О некорректно поставленных задачах // Матем. сб. 1963. Т. 61, № 2. С. 211—223.
 8. **Васин В. В., Еремин И. И.** Операторы и итерационные процессы фейеровского типа. Теория и приложения. Екатеринбург: УрО РАН, 2005.
 9. **Иванова К. Ф.** Заявка № 2014616687 "Программа для регуляризации некорректности физических и технических задач, позволяющая в значительной степени понизить экстремальную погрешность их решения (REGAS)".

K. F. Ivanova, Associate Professor, e-mail: klara.i2010@yandex.ru,
 Saint-Petersburg State University

The Approached Solutions of Ill-Conditioned SLAE with Inexact Factor

Estimations of a approached solutions of systems linear algebraic equations (SLAE) with ill-conditioned matrix and inexact coefficients and right-hand side are proposed. It is shown that estimation of solutions of the systems generated by "a sign" technique, are similar to the algebraic estimation for outer problem of the united set of solution interval systems linear algebraic equations (ISLAE) with interval matrixes. Association between occurrence of considerable errors of the solution linear systems by application of the "sign" technique initiating ill-conditioned matrixes, leading to an incorrect problem is considered. Regularization operators with numerical algorithm developed on the basis of A. N. Tikhonov's concept for a correct problems and construction of the approached steady solutions against small changes of the entrance data are used.

Keywords: linear algebraic equations, "sign" technique, interval system, regularization operators, incorrect problem, approached solution

References

1. **Ivanova K. F.** Znakovyy podhod k ocenke reshenija interval'nyh linejnyh sistem. *Informacionnye tehnologii*. 2012. N. 9. P. 46—52 (in Russian).
2. **Petrov Ju. P.** *Kak obespechit' nadezhnost' reshenija sistem uravnenij*. SPb.: BHV-SPb, 2009. 172 p. (in Russian).
3. **Sharyj S. P.** *Konechnomernyj interval'nyj analiz*. Novosibirsk: XYZ, 2007. 700 p. (in Russian).
4. **Tihonov A. N., Arsenin V. Ja.** *Metody reshenija nekorrektnykh zadach*. M.: Nauka. Glavnaja redakcija fiziko-matematicheskoy literatury, 1979. 284 p. (in Russian).
5. **Tihonov A. N.** *O reshenii nekorrektno postavlennykh zadach i metode reguljarizacii*. DAN SSSR. 1963. Vol. 151, N. 3. P. 501—504 (in Russian).
6. **Lavrent'ev M. M.** *O nekotorykh nekorrektnykh zadachah matematicheskoy fiziki*. Novosibirsk: Izd-vo SO AN SSSR, 1962. 92 p.
7. **Ivanov V. K.** O nekorrektno postavlennykh zadachah. *Matem. sb.* 1963. Vol. 61, N. 2. P. 211—223 (in Russian).
8. **Vasin V. V., Eremin I. I.** *Operatory i iteracionnye processy fejerovskogo tipa. Teorija i prilozhenija*. Ekaterinburg: Uro RAN, 2005 (in Russian).
9. **Ivanova K. F.** Zayavka № 2014616687 "Programma dlja reguljarizacii nekorrektnosti fizicheskikh i tehnicheskikh zadach, pozvoljajushhaja v znachitel'noj stepeni ponizit' jekstremal'nuju pogreshnost' ih reshenija (REGAS)" (in Russian).

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4
 Дизайнер Т. Н. Погорелова. Технический редактор Е. М. Патрушева. Корректор З. В. Наумова

Сдано в набор 05.12.2014 г. Подписано в печать 23.01.2014 г. Формат 60×88 1/8. Заказ PI215
 Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
 119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru